

Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального образования
Магнитогорский государственный технический университет им. Г.И.Носова

В.Е. Торчинский
В.Д. Тугарова
А.Н. Калитаев
Е.А. Ильина

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ C++

*Утверждено Редакционно-издательским советом университета
в качестве учебного пособия*

Магнитогорск
2009

УДК 681.3.06

Рецензенты:

Заведующий кафедрой прикладной информатики и управляющих систем
автоматики ГОУ ВПО ГТУ МИСис (Новотроицкий филиал),
доцент, кандидат технических наук

С.Н. Басков

Директор ООО «Магнитогорский научный информационно-технический
центр», профессор, доктор технических наук

А.М. Песин

Торчинский В.Е., Тутарова В.Д., Калитаев А.Н., Ильина Е.А.

Практикум по программированию на языке С++: учеб. пособие. -
Магнитогорск: ГОУ ВПО «МГТУ», 2009. - 108 с.

Пособие позволит приобрести представление об алгоритмах и конечном наборе базовых управляющих структур языка С++, о структуре программных средств, знания основ кодирования на языке высокого уровня С++. Изучение основ кодирования на языке С++ позволит использовать его как средство автоматизации обработки данных и решения проектных и научно-технических задач.

Предназначено для студентов специальностей 230105 - «Программное обеспечение вычислительной техники и автоматизированных систем» и 080116 - «Математические методы в экономике», изучающих дисциплины «Программирование на языке высокого уровня» и «Объектно-ориентированное программирование».

УДК 681.3.06

© ГОУ ВПО «МГТУ», 2009

© Торчинский В.Е., Тутарова В.Д.,
Калитаев А.Н., Ильина Е.А., 2009

Содержание

Введение	4
История развития языка C/C++	5
Этапы разработки программы и понятие алгоритма	8
Лабораторная работа № 1. Знакомство с компилятором C++.	
Создание консольных проектов и методов отладки программ	12
Лабораторная работа № 2. Программирование линейных алгоритмов...	19
Лабораторная работа № 3. Программирование алгоритмов ветвления...	33
Лабораторная работа № 4. Циклы	51
Цикл с предусловием (while)	51
Цикл с постусловием (do while)	52
Цикл с параметром (for)	53
Лабораторная работа № 5. Работа с одномерными массивами.....	66
Лабораторная работа № 6. Многомерные массивы	75
Лабораторная работа № 7. Структуры	86
Лабораторная работа № 8. Строки	93
Лабораторная работа № 9. Функции	99
Библиографический список	106
Приложение	107

ВВЕДЕНИЕ

В середине 1950-х годов, когда вычислительная техника прочно укоренилась в университетах и научно-исследовательских лабораториях США и Европы, наступило время стремительного прогресса в области программирования. С появлением языков высокого уровня программисты получили возможность больше времени уделять решению конкретной проблемы, не отвлекаясь особенно на весьма тонкие вопросы организации самого процесса выполнения задания на машине. Кроме того, появление этих языков ознаменовало первый шаг на пути создания программ, которые вышли за пределы научно-исследовательских лабораторий и финансовых отделов.

Языки программирования – это тщательно и изобретательно составленные последовательности слов, букв, чисел и mnemonicических сокращений, которыми люди пользуются для общения с компьютерами с целью выполнения задач. Каждый язык имеет свою грамматику и синтаксис. Больше чем за шестьдесят лет в мире программирования насчитывается несколько сотен таких языков. Языки программирования служат разнообразным целям – от решения сложных математических задач до создания музыкальной партитуры и машинной графики. Выбор языка программирования обычно определяется одним или, возможно, несколькими из трех факторов: язык должен быть удобен для программиста, пригоден для данного компьютера и для решения данной задачи.

Практикум по программированию на языке C++ содержит лабораторные работы, сгруппированные по темам для быстрого освоения данного языка, в которых содержится большое количество подробно разобранных задач и сопровождается заданиями для самостоятельной работы.

Кроме того, он может быть полезен для студентов, аспирантов и преподавателей.

ИСТОРИЯ РАЗВИТИЯ ЯЗЫКА C/C++

Язык Си называют языком структурного программирования благодаря структурированному потоку управления. Что имеется в виду? Существуют всего три основные структуры потока управления (поток управления можно определить как алгоритм перехода от текущего оператора к следующему):

Последовательная структура – следующим исполняется оператор, расположенный непосредственно после текущего.

Структура выбора – имеется несколько операторов; в зависимости от оценки некоторого выражения для исполнения выбирается только один из них, остальные игнорируются.

Структура повторения – текущий оператор исполняется снова и снова до тех пор, пока не будет удовлетворено некоторое условие завершения.

Из этих трех структур можно строить сколь угодно сложные управляющие конструкции, поскольку они подчиняются правилу суперпозиции, на место любого оператора некоторой структуры можно, в свою очередь, подставить любую структуру. При этом иногда последнюю требуется заключить в операторные скобки – в Си это фигурные скобки {} – подобно тому, как в арифметических выражениях используются обычные скобки. Любая последовательность операторов, заключенная в фигурные скобки, с точки зрения потока управления считается единым оператором.

Как известно, в свое время «большие» программы писались на языке Ассемблера, а это часто было неудобно и малопривлекательно. Поэтому у программистов появилось естественное желание как-то упростить, облегчить свою работу (ведь работа у них неимоверно тяжелая). Так появился язык программирования Си.

Среди современных языков программирования язык Си является одним из наиболее распространенных. Наряду с другими языками программирования он позволяет программисту ЭФФЕКТИВНО взаимодействовать с компьютером.

Язык Си универсален, однако наиболее эффективно его применение в задачах системного программирования – разработке трансляторов, операционных систем (как известно, операционная система UNIX была написана на Си), экранных интерфейсов, инструментальных средств.

Язык Си хорошо зарекомендовал себя эффективностью программ. Во многих случаях программы, написанные на языке Си, сравнимы по скорости с программами, написанными на языке Ассемблера, при этом они более наглядны и просты в сопровождении.

Одним из основных достоинств языка Си считается высокая переносимость написанных на нем программ между компьютерами с различной архитектурой, между различными операционными средами. Трансляторы языка Си существуют практически для всех используемых в настоящее время персональных компьютеров.

Язык Си был создан в 1972 году Денисом Ритчи, работавшим в компании Bell Telephone Laboratories, как язык для программирования в новой по тем временам операционной системе UNIX. Операционная система UNIX первоначально была написана на языке Ассемблера для мини-ЭВМ PDP-7 и перенесена затем на PDP-11.

На язык Си оказал значительное влияние его предшественник, язык Би, созданный Кеном Томпсоном, также принимавшим непосредственное участие в создании ОС UNIX. Язык Би, в свою очередь, являлся последователем языка BCPL, который был создан в 1969 году Мартином Ричардсом в рамках проекта «Комбинированный язык программирования» в Кембриджском университете в Лондоне. Вскоре система UNIX была переписана на языке Си, и в 1974–1975 годах ОС UNIX фирмы Bell Laboratories стала первым коммерческим продуктом, доказывающим, что операционная система может быть успешно написана на языке высокого уровня, если этот язык является достаточно мощным и гибким.

В 1978 году Бра Керниган и Деннис Ритчи написали книгу «Язык программирования Си». Эта работа, которая среди программистов называется «белой книгой», «большой голубой Си» и "K&R", стала стандартом описания языка Си.

К сожалению, описание языка не было строгим и полным и содержало ряд моментов, допускающих неоднозначное толкование. Это привело к тому, что в последующем различные разработчики систем программирования трактовали язык Си по-разному. В течение долгого времени фактическим стандартом языка Си служила его реализация в седьмой версии операционной системы UNIX, однако заявления о совместимости некоторой системы программирования с реализацией языка Си в операционной системе UNIX, как правило, не означали точного соответствия.

Для исправления этой ситуации в 1983 году при Американском Национальном Институте Стандартов (ANSI) был образован комитет по стандартизации языка Си. В октябре 1986 года разработанный этим комитетом проект был опубликован для общественного обсуждения. В 1989 г. окончательный вариант проекта был утвержден в качестве стандарта ANSI. На момент создания "K&R" существовали компиляторы языка Си для ЭВМ PDP-11, Interdata8/32, Honeywell 6000 и IBM 370. В дальнейшем список машин был продолжен.

Это естественно, поскольку язык Си настолько гибок и независим от конкретной аппаратной конфигурации, что может быть использован для решения практически любых задач. Стив Шастак в своей книге «Вариации в Си» приводит яркий пример на эту тему: «Некоторые программы, написанные на языке Си, успешно функционируют в окружении, в котором нет ни терминалов, ни принтеров, ни дисководов, и даже... никакой операционной системы. Это может быть, например, программа, управляющая работой станка или, скажем, лифта. Правда, в ней мало

будут использоваться библиотечные функции и функции стандартного ввода / вывода (и многие другие) и «общаться» такая программа будет только с сенсорами или кнопками, но это уже детали».

За исключением незначительных деталей, С – это подмножество языка Си++. С++ – это язык программирования общего назначения, его предназначение – сделать работу серьезных программистов более приятным занятием. Задание новых общеупотребительных и прикладных типов – наиболее фундаментальная программистская деятельность в С++.

С++, очевидно, больше всего заимствовал из языка Си. Вторым источником вдохновения был язык Simula67. Само название С++ появилось летом 1983 г. Название С++ придумал Рикк Маскитти. Как известно, “++” – это оператор инкрементации в языке Си.

По мере распространения С++ и роста его возможностей, вновь и вновь поднимался вопрос, нужно ли сохранять совместимость с языком Си. После многочисленных споров было пересмотрено определение С++ с тем, чтобы гарантировать, что любой конструктор, правильный и в Си, и в С++, имеет одно и то же значение в обоих языках.

Особенности языка Си

Язык Си имеет ряд существенных особенностей, которые выделяют его среди других языков программирования. В значительной степени на формирование идеологии языка повлияла цель, которую ставили перед собой его создатели, – обеспечение системного программиста удобным инструментальным языком, который мог бы заменить язык Ассемблера. В результате появился язык программирования высокого уровня, обеспечивающий необычайно легкий доступ к аппаратным средствам компьютера.

Иногда Си называют языком «среднего» уровня. С одной стороны, как и другие современные языки высокого уровня, язык Си поддерживает полный набор конструкций структурного программирования, модульность, блочную структуру программ, раздельную компиляцию. С другой стороны, язык Си имеет ряд низкоуровневых черт.

Вот некоторые характерные для Си особенности:

- в языке реализованы некоторые операции низкого уровня (над битами и т. п.);

- базовые типы языка отражают те же объекты, с которыми приходится иметь дело в программе на языке Ассемблера, – байты, машинные слова, символы, строки;

- язык Си поддерживает механизм указателей на переменные и функции, арифметику указателей;

- как никакой другой язык программирования высокого уровня, язык Си «доверяет» программисту. Даже в таком существенном вопросе, как преобразование типов данных, налагаются незначительные ограничения.

ЭТАПЫ РАЗРАБОТКИ ПРОГРАММЫ И ПОНЯТИЕ АЛГОРИТМА

В процессе создания любой программы, будь то небольшая учебная программа или серьезный проект, над которым работают десятки (а то и сотни) программистов, можно выделить несколько этапов. Затраты труда и времени на их выполнение различны, различаются эти затраты и для разных программ. Некоторые из этапов могут быть опущены или пройдены «незаметно», однако анализ процесса разработки приводит к выводу о том, что почти всегда, явно или неявно, приходится проходить следующие этапы разработки программы:

1. *Постановка задачи.* Постановка задачи выполняется заказчиком, в качестве которого может выступать внешняя организация, организация, в которой работает программист, начальник программиста, преподаватель, сам программист. На этом этапе задача, которую необходимо решить посредством составления программы для компьютера, формулируется на естественном языке. При этом важно осознать, что решение данной задачи с помощью компьютера – действительно оптимальный способ получения результата. Вряд ли стоит составлять программу для вычисления суммы двух чисел – в этом случае достаточно воспользоваться калькулятором.

2. *Анализ, формализованное описание задачи, выбор модели.* Выбор или разработка алгоритма решения задачи. Анализ задачи включает определение входных и выходных данных, выявление возможных ограничений на их значения и обычно завершается формализованным описанием задачи, которое часто предполагает ее математическую формулировку. Если речь идет о моделировании каких-либо явлений или процессов, на этом этапе разрабатывается математическая модель процесса (явления). В этом случае определяются факторы, которые играют основную роль, и отбрасываются те факторы, действие которых незначительно и которыми поэтому можно пренебречь. Этот этап можно разбить на следующие шаги:

Шаг 1. Математическая постановка задачи:

- что дано – перечисление исходных данных;
- что требуется – перечисление результатов;
- условия допустимости исходных данных.

Шаг 2. Математическая модель – все необходимые для получения результатов правила и законы.

Шаг 3. Метод решения – оптимальное использование имеющейся в распоряжении математической модели. Выбор или разработка алгоритма и численного метода решения задачи имеют важнейшее значение для успешной работы над программой. Тщательно проработанный алгоритм решения задачи – необходимое условие эффективной работы по составлению программы.

3. *Проектирование общей структуры программы.* На этом этапе происходит «архитектурная» проработка проекта. Определяются те части алгоритма, которые целесообразно оформить в виде подпрограмм, модулей. Определяется и способ хранения информации – в виде набора простых переменных, массивов или других структур.

4. *Кодирование.* Это запись алгоритма на языке программирования. Если алгоритм решения задачи, структура программы и структура данных тщательно продуманы и аккуратно записаны, затраты времени на кодирование уменьшаются, а вероятность ошибок на этом этапе снижается.

5. *Отладка и верификация программы.* Отладка и верификация программы представляют собой очень важную часть процесса разработки программы. Отладка заключается в устранении ошибок программирования, ошибок перевода алгоритма на язык программирования. Верификация – это доказательство того, что программа работает «правильно», дает правильный результат. Для этого разрабатывается система тестов, которые могут представлять собой специально подобранные наборы параметров, для которых задача может быть решена точно. Это могут быть, например, какие-нибудь предельные случаи. Если результат, полученный с помощью программы, совпадает (с учетом погрешности машинного счета) с ожидаемым, есть основание полагать, что программа работает корректно. Но это всего лишь основание, а не абсолютная уверенность! Среди начинающих программистов распространено убеждение, что если программа успешно откомпилирована и, будучи запущена на выполнение, выдает на экран ряды цифр, задача решена. На самом же деле программа готова, если разработчик смог доказать заказчику (да и самому себе), что результат работы программы является решением поставленной задачи.

6. *Получение результата, его интерпретация и, возможно, последующая модификация модели.* После проверки программы и устранения ошибок необходимо проанализировать полученный результат. Если речь идет о моделировании какого-то природного процесса, следует сравнить полученные с помощью компьютера результаты и результаты наблюдений. Процесс такого анализа называется интерпретацией результатов расчета.

7. *Публикация или передача заказчику результата работы.* В научных исследованиях значение имеют результаты моделирования, которые публикуются в научных журналах. В других случаях конечным результатом работы может быть сама программа, которая передается заказчику для дальнейшей эксплуатации.

8. *Сопровождение программы.* Сопровождение программы предполагает консультации заказчика по работе программы, устранение замеченных в процессе ее эксплуатации недостатков (а возможно, и ошибок), обучение пользователей работе с программой. Этот, заключительный этап имеет особое значение для больших и сложных программ.

Алгоритмом называется конечный набор правил, расположенных в определенном логическом порядке, которые необходимо выполнить для решения задачи. Практически решение любой задачи требует получения результата по заданным исходным данным. То есть можно сказать, что алгоритм описывает последовательный процесс преобразования исходных данных в результат.

Разработать алгоритм решения задачи означает разбить задачу на последовательно выполняемые шаги (этапы), причем результаты выполнения последующих этапов могут использоваться при выполнении последующих. При этом должны быть четко указаны как содержание каждого этапа, так и порядок выполнения этапов. Отдельный этап (шаг) алгоритма представляет собой либо другую, более простую задачу, алгоритм решения которой разработан ранее, либо должен быть достаточно простым и понятным без пояснений.

Четко сформулированная последовательность правил, описывающих этот процесс, и является *алгоритмом*.

Алгоритм обладает следующими основными свойствами, раскрывающими его определение:

1. *Дискретность*. Это свойство состоит в том, что алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов). При этом для выполнения каждого шага (этапа) алгоритма требуется некоторый конечный отрезок времени. То есть преобразование исходных данных в результат осуществляется во времени дискретно.

2. *Определенность* (или детерминированность). Это свойство состоит в том, что каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

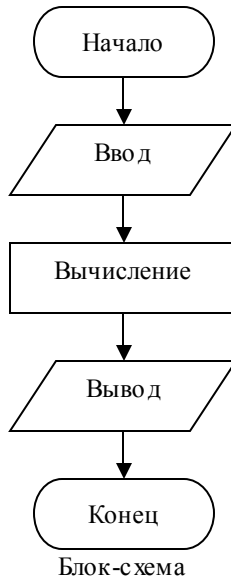
3. *Результативность* (или конечность). Это свойство состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов.

4. *Массовость*. Это свойство состоит в том, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Любой алгоритм можно представить в виде трех основных блоков:

1. Ввод исходных данных.
2. Обработка (или вычисление) данных.
3. Вывод (или представление) обработанных данных.

При этом любой из этапов может отсутствовать, но большинство всех задач подчиняется именно этой схеме (см. рисунок).



Блок-схема позволяет наглядно представить структуру алгоритма. В настоящее время существует технология разработки программ без использования схем. Однако независимо от этого на начальном этапе изучения программирования рекомендуется использовать блок-схемы. Использование схем обеспечивает приобретение прочных навыков разработки алгоритмов.

Основными объектами программирования являются переменные. Переменные в программе отличаются от переменных, используемых в записи математических формул. Несмотря на сходство терминов, правила использования переменных в программах для компьютера отличаются от правил работы с математическими переменными. В программировании переменную можно трактовать как область оперативной памяти компьютера, которой присвоено определенное имя. Содержимое этих ячеек может меняться, но имя переменной остается неизменным. В математике значение переменной в рамках определенной задачи неизменно, но меняется в других задачах из данного класса.

Таким образом, при составлении плана решения задач руководствуются следующими этапами:

1. Формулировка задачи.
2. Математическая постановка задачи.
3. Выбор переменных программы.
4. Схема алгоритма.
5. Программа.
6. Тесты.
7. Результаты тестирования.

Лабораторная работа № 1 ЗНАКОМСТВО С КОМПИЛЯТОРОМ C++. СОЗДАНИЕ КОНСОЛЬНЫХ ПРОЕКТОВ И МЕТОДОВ ОТЛАДКИ ПРОГРАММ

Вызов C++ осуществляется или через меню

Пуск→Программы→Borland Developer Studio 2006→Turbo C++, или щелчком мышью по пиктограмме с соответствующим именем, если она размещена на рабочем столе. После запуска Turbo C++ появляется главное окно программы, представленное на рис. 1.1. Рабочий стол Borland Developer Studio 2006 включает в себя четыре окна:

- окно *Project Workspace 1* (окно рабочей области) предназначено для оказания помощи при написании и сопровождении больших много-файловых программ, одна из вкладок этого окна содержит список файлов проекта;

- окно *Editor 2* (окно редактора) используется для ввода и проверки исходного кода;

- окно *Output 3* (окно вывода) служит для вывода сообщений о ходе компиляции, сборки и выполнения программы (сообщения о возникающих ошибках появляются именно в этом окне);

- окно *Output 3* (окно вывода) служит для вывода сообщений о ходе компиляции, сборки и выполнения программы (сообщения о возникающих ошибках появляются именно в этом окне).

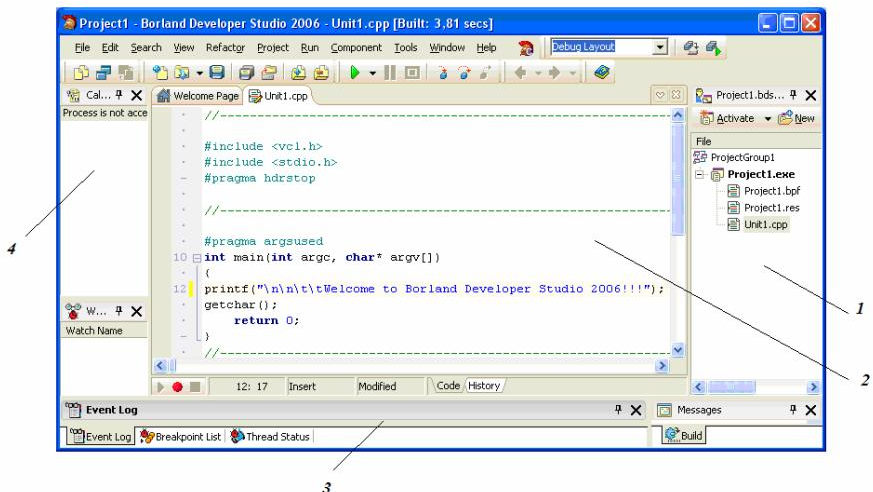


Рис. 1.1

Под заголовком главного окна, как и во всех Windows-приложениях, находится строка меню. Для кнопок панелей инструментов предусмотрена удобная контекстная помощь (всплывает подсказка с назначением кнопки). Borland Developer Studio 2006 позволяет строить проекты разных типов, ориентированных на различные сферы применения. Так как этот пакет спроектирован на Windows-платформе, то почти все типы проектов являются оконными Windows-приложениями с соответствующим графическим интерфейсом. В то же время разработчики предусмотрели работу и с так называемыми консольными приложениями.

Для создания нового проекта в режиме консоли выполняем команду *File->New->Other...* (рис. 1.2).

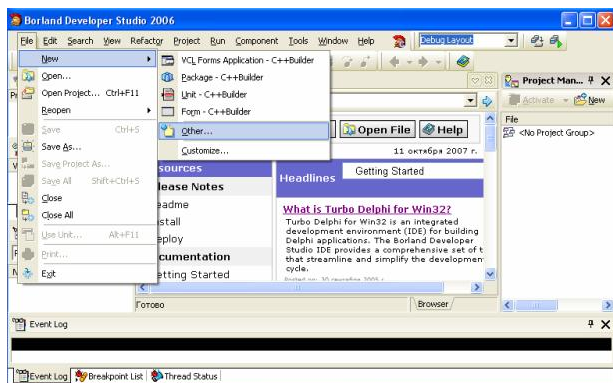


Рис. 1.2

В открывшемся диалоговом окне *New* выберите тип *Console Application* (рис. 1.3), а затем в окне *New Console Application* отредактируйте параметры работы с консольным приложением, как рекомендуется на рис. 1.4.

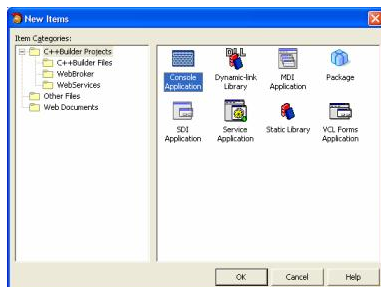


Рис. 1.3

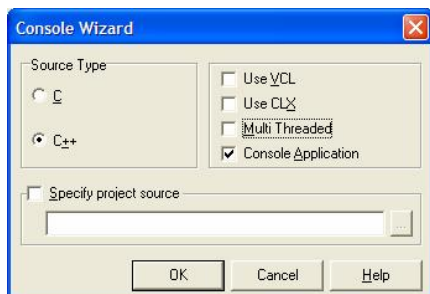


Рис. 1.4

При запуске консольного приложения операционная система создает консольное окно, через которое происходит весь ввод-вывод программы. Внешне это напоминает работу в операционной среде MS-DOS в режиме командной строки. Этот тип приложений удобен при изучении языка C/C++.

До начала написания программного кода с целью дальнейшего предотвращения потерь данных проекту необходимо присвоить имя с помощью команды *File->Save All* (рис. 1.5).

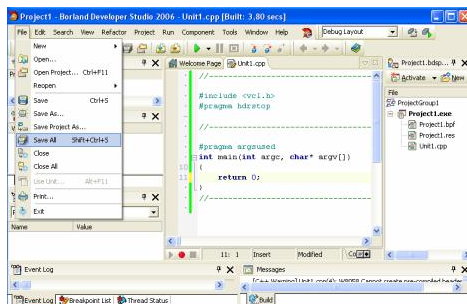


Рис. 1.5

Затем выбираем папку, предварительно созданную для хранения разработанных программ, и в строке Имя файла последовательно вводим имя модуля и имя проекта (рис. 1.6), заканчивая каждую операцию нажатием кнопки «Сохранить».

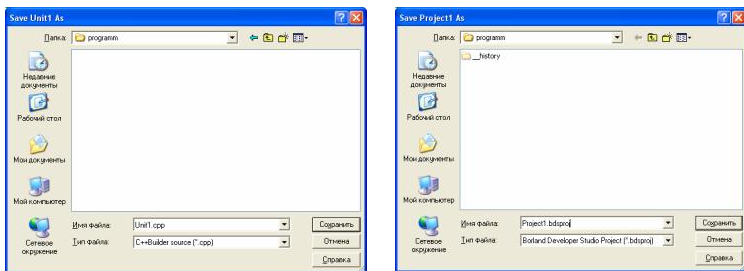


Рис. 1.6

Операция открытия проекта производится выполнением команды *Open Project* (рис. 1.7).

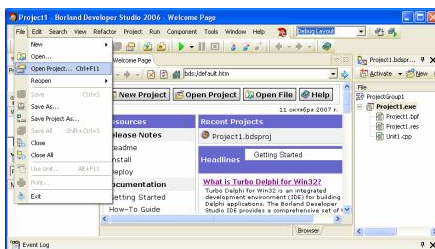


Рис. 1.7

Затем необходимо выбрать папку, в которой хранится проект и сам файл проекта (рис. 1.8) и нажать кнопку «Открыть».

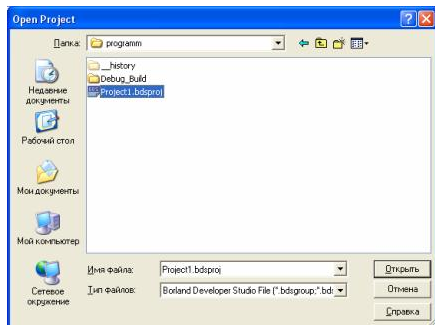


Рис. 1.8

В результате работы над программой часто возникает необходимость просмотреть промежуточные результаты вычисления. Это достигается с помощью окна отладки *Watch*. Рассмотрим работу программы на примере нахождения суммы и разности двух чисел, код которой приведен ниже.

```
int main()
```

```

{
    int a=5, b=2, c, d;
    c=a+b;
    d=a-b;
    return 0;
}

```

Для вызова окна *Watch* выполним команду *Run->Add Watch*. В верхней строке *Expression* вводим имена переменных, работу которых необходимо проконтролировать (рис. 1.9).

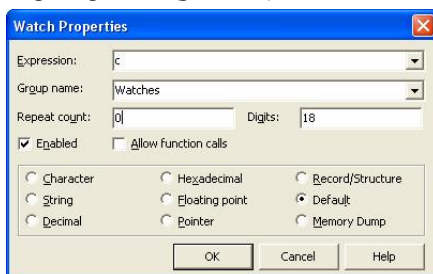


Рис. 1.9

В этом случае помимо экранной подсказки, переменные отображаются в окне *Watch List*, расположенном в левой части окна (рис. 1.10). В этом окне приведены значения переменных, при этом выбор переменных для просмотра определяет сам создатель программы.

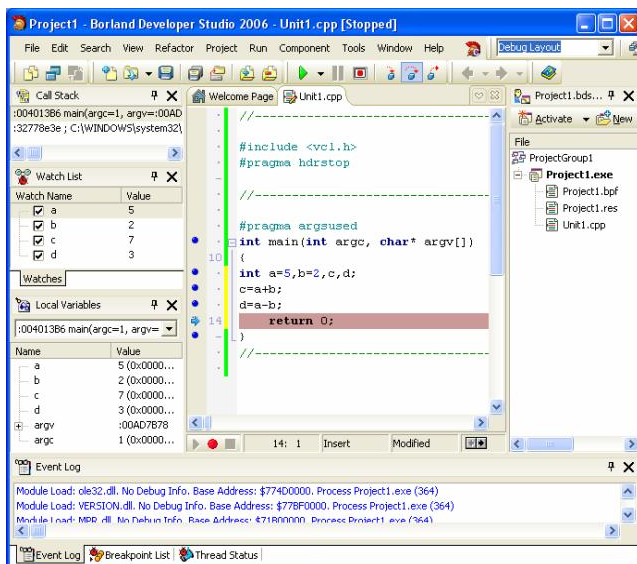


Рис. 1.10

Точка прерывания позволяет остановить выполнение программы перед любой выполняемой инструкцией (оператором) с тем, чтобы продолжать выполнение программы либо в пошаговом режиме, либо в непрерывном режиме до следующей точки прерывания. Точка прерывания задается щелчком мыши перед некоторым оператором и обозначается в виде красного кружочка на левом поле окна редактирования. Повторный щелчок на указанной метке снимает точку прерывания. В программе может быть несколько точек прерывания. Результаты работы программы с использованием окна Watch приведены на рис. 1.11.

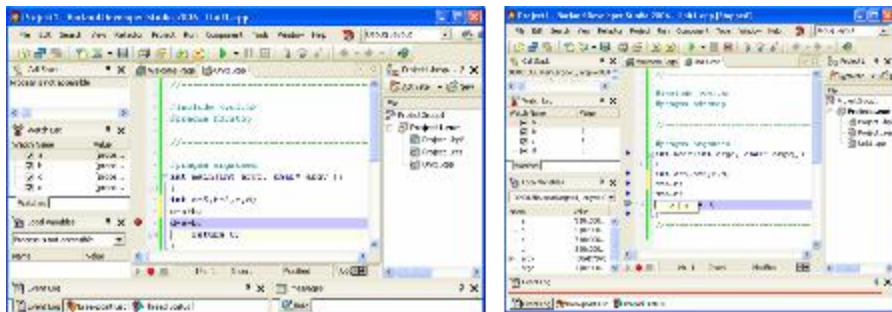


Рис. 1.11

На рис. 1.12 представлен исходный код программы, выполняющей вывод текстовой информации ("Welcome to Borland Developer Studio 2006!!!") на экран консольного приложения, при этом используем функцию вывода *printf*.

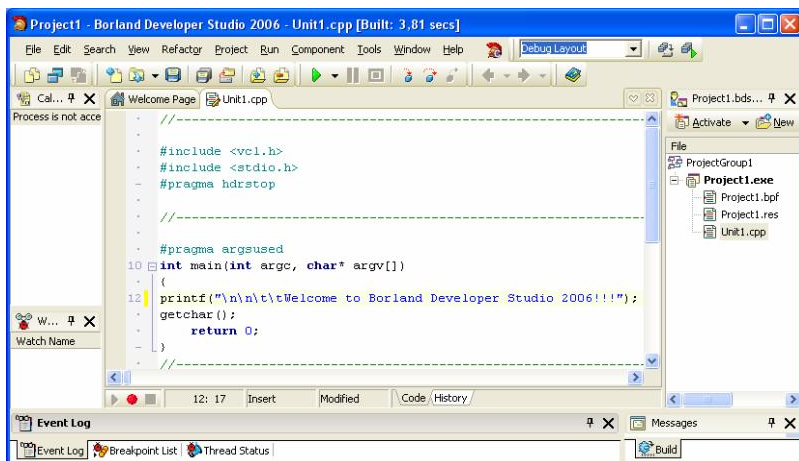


Рис. 1.12

Для компиляции, компоновки и запуска программы необходимо выполнить команду *Run->Run* или нажать клавишу *F9*.

Результат работы программы приведен на рис. 1.13.

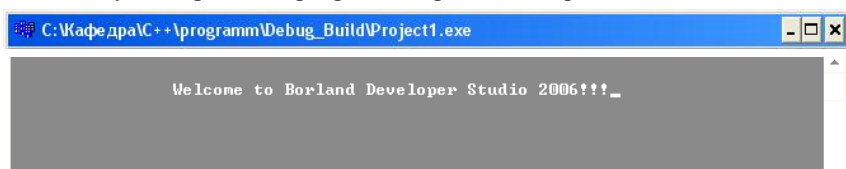


Рис. 1.13

Лабораторная работа № 2

ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

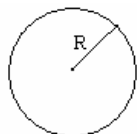
Если в программе все операторы выполняются последовательно, один за другим, такая программа называется линейной. Рассмотрим ряд задач с использованием линейных алгоритмов.

Задача 2.1

1. Формулировка задачи.

Требуется написать программу, которая по заданному радиусу окружности вычисляет: длину окружности, площадь круга, площадь сферы и объем шара.

2. Математическая постановка задачи.



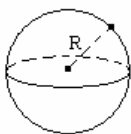
Для расчета перечисленных характеристик воспользуемся формулами:

длина окружности – $C = 2\pi R$;

площадь круга – $S = \pi R^2$;

площадь сферы – $S = 4\pi R^2$;

объем шара – $V = \frac{4}{3}\pi R^3$.



3. Выбор переменных программы.

Из приведенного выше решения определяем следующие переменные:

исходные данные – радиус окружности (R);

справочные данные – число π (Pi);

результат – длина окружности (C), площадь круга (Scr), площадь сферы (Scф) и объем шара (V).

Так как радиус окружности может принимать любые значения (2; 2,5; 3,75), все переменные определяем как действительные числа.

4. Схема алгоритма.

Составим блок-схему этой задачи (рис. 2.1).

5. Программа.

Программа на C++ состоит из функций. Функция – это поименованная последовательность операторов. Функция состоит из заголовка и тела. При создании модуля автоматически создается заголовок главной функции программы `int main()`. Имя `main` указывает на то, что именно с этой функции начинается выполнение программы. Обычно главная функция возвращает целочисленное значение, за это отвечает оператор `return 0`.

В фигурных скобках (`{ ... }`) записывается тело функции.

```
#include <stdio.h>
```

```
#include <math.h>
```

```

int main()
{
    float R,C,S1,S2,V;
    printf("Input R->");
    scanf("%f",&R);
    C=2*M_PI*R;
    S1=M_PI*pow(R,2);
    S2=4*M_PI*pow(R,2);
    V=4*M_PI*pow(R,3)/3.0;
    printf("\nC=%6.2f",C);
    printf("\nS1=%6.2f",S1);
    printf("\nS2=%6.2f",S2);
    printf("\nV=%6.2f",V);
    getchar(); getchar();
    return 0;
}

```

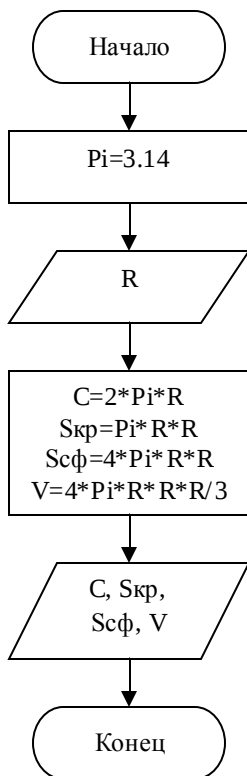


Рис. 2.1

Внесем пояснения по тексту программы. Так как при выборе переменных все они (*R*, *C*, *S1*, *S2*, *V*) были определены как действительные числа, то им присвоен вещественный тип *float*. В общем случае тип переменных выбирается исходя из возможного диапазона значений и требуемой точности представления данных. В табл. 2.1 приведено описание для переменных, представляющих число целого типа.

Таблица 2.1

Встроенные типы данных

<i>Тип данных</i>	<i>Размер (бит)</i>	<i>Диапазон</i>
char	8	-128 – 127
signed char	8	-128 – 127
unsigned char	8	0 – 255
short	16	-32768 – 32767
unsigned short	16	0 – 65535
int	Соответствует разрядности ОС	
unsigned int	Соответствует разрядности ОС	
long	32	-2147483648 – 2147483647
unsigned long	32	0 – 4294967295
float	32	$3.4 \cdot 10^{-38}$ – $3.4 \cdot 10^{38}$ (7 цифр)
double	64	$1.7 \cdot 10^{-308}$ – $1.7 \cdot 10^{308}$ (15 цифр)

Синтаксис оператора объявления можно описать примерно так:

тип имя_переменной [= инициализирующее_значение][, ...];

Как и любой другой оператор C, он оканчивается точкой с запятой.

В данном варианте программы использован ввод-вывод с помощью функций библиотеки *<stdio.h>* (в стиле C). Функция *printf("Input R->")* выполняет вывод переданного ей в качестве параметра строкового литерала *Input R->*, то есть последовательности символов в кавычках, на стандартное устройство вывода (дисплей). Символ '\n' называется управляющей последовательностью и задает переход на новую строку.

Для ввода исходных данных используется функция *scanf("%f",&R)*. Параметры любой функции перечисляются через запятую. В первом параметре функции *scanf* в виде строкового литерала *%f* задается спецификация формата вводимой величины, соответствующая типу вводимой

переменной. Спецификация `%f` соответствует типу `float`. В качестве второго параметра функции передается адрес переменной, по которому будет помещено вводимое значение. Операция взятия адреса обозначается `&`.

Для вывода результата используем оператор `printf("\nC=%6.2f",C)`. Все символы литерала, за исключением управляющей последовательности `\n` и спецификации формата `%f`, выводятся на дисплей без изменения. При выводе форматная спецификация будет заменена конкретным значением переменной `C`.

Формат вывода чисел можно уточнить при помощи модификаторов формата – чисел, которые записаны перед спецификацией. Первое число «6» задает минимальное количество позиций, отводимых под выводимую величину, второе «2» – сколько из этих позиций отводится под дробную часть величины. Необходимо учитывать, что точка тоже занимает одну позицию. Если заданного количества окажется недостаточно, то компилятор автоматически выделит поле необходимой длины.

Функции ввода-вывода данного типа `printf`, `scanf` и `getchar()` без подключения библиотеки `<stdio.h>` работать не будут.

Существует другой способ организации ввода-вывода с помощью классов (в стиле C++). Если требуется ввести с клавиатуры данные, применяется так называемое приглашение к вводу `cout<<"Input R->"<<endl`. На экран выводится указанная в операторе строка символов `Input R->`, и курсор переводится на новую строку с помощью манипулятора `endl`.

В операторе `cin>>R` выполняется ввод с клавиатуры одного числа в переменную `R`. Для этого используется стандартный объект `cin` и операция извлечения (чтения) `>>`.

Для вывода результата используется объект `cout<<"C="<<C<<endl`, при этом все символы, находящиеся внутри кавычек, включая пробелы, выводятся без изменения. При выводе значений переменных выполняется преобразование из внутреннего представления числа в строку символов, представляющих это число. Под значение отводится ровно столько позиций, сколько необходимо для вывода всех его значащих цифр. В этом случае подключается библиотека `<iostream.h>`.

```
#include<iostream.h>
#include <math.h>
int main()
{
    float R,C,S1,S2,V;
    cout<<"Input R->"<< endl;
    cin>>R;
    C=2*M_PI*R;
    S1=M_PI*pow(R,2);
    S2=4*M_PI*pow(R,2);
```

```

V=4*M_PI* pow(R,3)/3.0;
cout<<"C="<<C<<endl;
cout<<"S1="<<S1<<endl;
cout<<"S2="<<S2<<endl;
cout<<"V="<<V<<endl;
getchar(); getchar();
return 0;
}

```

Так как программы содержат константу M_PI , то необходимо подключить библиотеку `<math.h>`. Список математических функций библиотеки `<math.h>` приведен в табл. 2.2.

Таблица 2.2

Список математических функций библиотеки `<math.h>`

Функция	Описание
acos	Возвращает арккосинус аргумента
asin	Возвращает арксинус аргумента
atan	Возвращает арктангенс аргумента
atan2	Возвращает арктангенс отношения аргументов
ceil	Округляет вверх
cos	Вычисляет косинус
cosh	Вычисляет гиперболический косинус
exp	Возвращает степень числа e
fabs	Возвращает модуль числа
floor	Округляет вниз
fmod	Возвращает остаток от деления x на y
frexp	Выделяет из числа мантиссу и экспоненциальную часть
ldexp	Преобразует мантиссу и показатель степени в число
log	Вычисляет натуральный логарифм
log10	Вычисляет логарифм по основанию 10
modf	Разбивает число на целую и дробную части
pow	Возводит число в степень
sin	Вычисляет синус
sinh	Вычисляет гиперболический синус
sqrt	Вычисляет квадратный корень
tan	Вычисляет тангенс аргумента
tanh	Вычисляет гиперболический тангенс аргумента

Операции языка C/C++ приведены в табл. 2.3.

Таблица 2.3

Операции языка C/C++

Операция	Описание	Приоритет	Ассоциация
Первичные и постфиксные операции			
[]	вычисление индексного выражения	16	слева направо
()	вызов функции	16	слева направо
.	элемент структуры	16	слева направо
->	элемент указателя	16	слева направо
++	постфиксный инкремент	15	слева направо
--	постфиксный декремент	15	слева направо
Одноместные операции			
++	префиксный инкремент	14	справа налево
--	префиксный декремент	14	справа налево
sizeof	размер в байтах	14	справа налево
(тип)	приведение типа	14	справа налево
~	поразрядное NOT	14	справа налево
!	логическое NOT	14	справа налево
-	унарный минус	14	справа налево
&	взятие адреса	14	справа налево
*	разыменование указателя	14	справа налево
Двухместные и трехместные операции			
<i>Мультипликативные</i>			
*	умножение	13	слева направо
/	деление	13	слева направо
%	взятие по модулю	13	слева направо
<i>Аддитивные</i>			
+	сложение	12	слева направо
-	вычитание	12	слева направо
<i>Поразрядного сдвига</i>			
<<	сдвиг влево	11	слева направо
>>	сдвиг вправо	11	слева направо

Продолжение табл. 2.3

Операция	Описание	Приоритет	Ассоциация
<i>Отношения</i>			
<	меньше	10	слева направо
<=	меньше или равно	10	слева направо
>	больше	10	слева направо
>=	больше или равно	10	слева направо
==	равно	9	слева направо
!=	не равно	9	слева направо
<i>Поразрядные</i>			
&	поразрядное AND	8	слева направо
^	поразрядное XOR	7	слева направо
	поразрядное OR	6	слева направо
<i>Логические</i>			
&&	логическое AND	5	слева направо
	логическое OR	4	слева направо
<i>Условные</i>			
? :	условная операция	3	справа налево
<i>Присваивания</i>			
=	присваивание	2	справа налево
*=	присвоение произведения	2	справа налево
/=	присвоение частного	2	справа налево
%=	присвоение модуля	2	справа налево
+=	присвоение суммы	2	справа налево
-=	присвоение разности	2	справа налево
<<=	присвоение левого сдвига	2	справа налево
>>=	присвоение правого сдвига	2	справа налево
&=	присвоение AND	2	справа налево

Окончание табл. 2.3

Операция	Описание	Приоритет	Ассоциация
$\wedge=$	присвоение XOR	2	справа налево
$ =$	присвоение OR	2	справа налево
,	последовательного вычисления	1	слева направо

6. Тесты.

Для испытания программы можно использовать тесты, представленные в табл.2.4.

Таблица 2.4

Тесты для задачи 2.1

<i>R</i>	<i>C</i>	<i>Skp</i>	<i>Scf</i>	<i>V</i>
0	0	0	0	0
2	12,57	12,57	50,27	33,51
2,25	14,14	15,9	63,62	47,71

7. Результаты тестирования.

Результат работы программы представлен на рис.2.2.

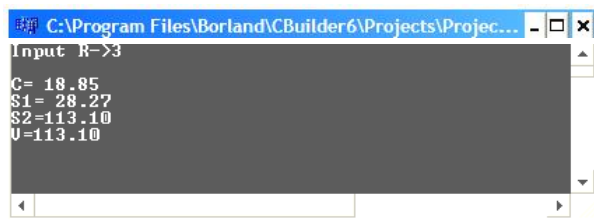


Рис. 2.2

Задача 2.2

Составить программу вычисления значения функции для любого заданного x :

$$y = \frac{1 + \sin^3(x^2 + 2x)}{2 + \left| x - \frac{2x}{\sin x + 2} \right|} + e^{2x+1}.$$

Листинг программы

```
#include <stdio.h>
#include <math.h>
int main()
{
```

```

float x,x1,x2,y;
printf("Input x->");
scanf("%f",&x);
x1=pow(sin(pow(x,2)+2*x),3)+1;
x2=fabs(x-2*x/(sin(x)+2))+2;
y=x1/x2+exp(2*x+1);
printf("\ny=%6.2f",y);
getchar(); getchar();
return 0;
}

```

Результаты тестирования.

Результат работы программы представлен на рис.2.3.

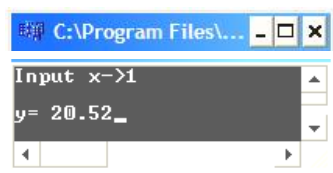


Рис.2.3

Задача 2.3

Треугольник задается координатами своих вершин на плоскости: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Требуется написать программу, которая вычисляет площадь треугольника ABC.

Для решения задачи можно использовать формулу Герона:

$S = \sqrt{p(p-a)(p-b)(p-c)}$, где p – полупериметр; a, b, c – длины сторон треугольника, которые вычисляются согласно рис.2.4.

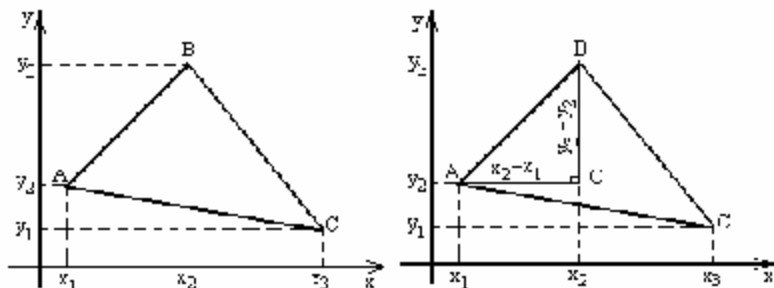


Рис. 2.4

Задача 2.4

Полторы кошки съедают полторы мышек за полтора часа. Требуется написать программу, которая определяет время, за которое n кошек съедят m мышек?

Задача 2.5

Требуется написать программу, которая по введенному количеству секунд определяет количество суток, часов, минут и секунд. Например, 4000 с = 0 сут 1 ч 6 мин 40 с.

Задача 2.6

Требуется написать программу, которая по введенному времени определяет меньший угол между часовой и минутной стрелкой. Например,

$$12 \text{ ч } 00 \text{ мин} - \text{угол} = 0.$$

$$2 \text{ ч } 30 \text{ мин} - \text{угол} = 105.$$

Индивидуальные задания

по теме «Программирование линейных алгоритмов»

Задание И2.1. Составить программу для вычисления значения функции для любого заданного x (табл.2.4). При выводе исходных данных результат округлять до второго знака после запятой.

Таблица 2.4

Условия вычисление значения функций

№	Значение функции	№	Значение функции
1	$y = x \ln(x^4 + 5) + \frac{x}{\cos x + 2} + 2 \sin \sqrt{x^2 + 1}$	16	$y = \frac{x^2 - 7x + 10}{x^2 + 8x + 12} - \ln(\cos x + 2) + e^{-2x}$
2	$y = \frac{\sin x + \cos x}{\sqrt[3]{\cos x - \sin x + 3}} + (e^x + 1)^2$	17	$y = x \ln(x^2 + 1) + \frac{x}{\cos^2 x + 1} + \sqrt{ x + 1 }$
3	$y = \frac{\cos x}{\pi + 2x^2} + x \ln(x^4 + \sqrt{ x })$	18	$y = \sin \sqrt{x^2 + 1} - \frac{\sin \sqrt{ x - 1 }}{\ln(x^2 + 1)}$
4	$y = \frac{1 + \sin \sqrt{ x + 1 }}{\cos^2(12x^2 + 4) + 1} + e^{\frac{\sqrt{ x }}{\sin^2 x + 2}}$	19	$y = e^{\sin^2 x + 1} - \frac{x}{\ln \cos^3 x + 5 } + (1 + x)^x$
5	$y = x + \frac{x^3}{3} - \frac{x^5}{5} + \frac{xe^{2x}}{\sin^2 x + 1}$	20	$y = 3^x + 4x - \frac{\ln\left \frac{\cos^2 x + 1}{x^2 + 1}\right }{2 \sin x + 25}$
6	$y = \frac{\ln \cos x }{\ln(1 + x^2)} + e^{2^x} + 10^{\sin x}$	21	$y = x - \frac{10 \sin x}{\ln\left(\frac{e^{-2x}}{2 \sin^2 x + 1}\right)} + x^4 - x^5 $
7	$y = 2^{-x} - \cos x + \frac{\sin(2x)}{\ln\left(\cos \frac{e^{-x}}{x^2 + 1} + 2\right)}$	22	$y = \frac{1 + \sin^3(x^2 + 2x)}{2 + \left x - \frac{2x}{\sin x + 2}\right } + e^{2x+1}$
8	$y = 2 \ln \left \frac{\sin x + 2}{\cos^2 x + 1} \right + e^{\sin^2 x + 1}$	23	$y = \sqrt{2 \sin^3 3x} - \frac{\ln(\cos^2 x + 1)}{\ln x^4 + 2x^2 + 10 } - e^{\frac{-2x}{5}}$

Окончание табл.2.4

№	Значение функции	№	Значение функции
9	$y = x^2 - x^3 - \frac{7x}{(x^3 + 15x)^2 + 1} + e^{\cos^2 x + 1}$	24	$y = \frac{\cos^2 x}{\sin x + 2} - x \sin x + \frac{a + \sin^2(x^2 + 2)}{\left x - \frac{1}{\sin^2 x + 1}\right }$
10	$y = x \ln(x^2 + 1) + \frac{x}{\cos^2 x + 1} + \sqrt{ x + 1 }$	25	$y = \cos^2 \sin \frac{1}{x^2 + 1} - \frac{\ln \left \frac{\cos x + 2}{\sin^2 x + 1,2} \right }{e^{-2x} + 1}$
11	$y = \sin \sqrt{x^2 + 1} - \frac{\sin \sqrt{ x - 1 }}{\ln(x^2 + 1)}$	26	$y = \frac{1 + \sin^2(x^2 + 2x)}{2 + \left x - \frac{2x}{(1 + x^2)(\sin x + 2)}\right } + \ln \sin^2 x - 2 $
12	$y = e^{-2x+1} + \frac{x^2 + 12x - 3}{\ln \sin^3 x + 1 }$	27	$y = \frac{(\sin^2 x + \cos x)^{3x}}{\ln \left(\frac{\cos x + 2}{e^{\sin^2 x + 1}} \right)} - (5 + \cos^2 x)^x$
13	$y = \frac{1 + \sin \sqrt{x^2 + 1}}{\cos^2 12x + 1} + \ln(1 + x^2)$	28	$y = x - 10^{\sin x} - \frac{\ln \cos x + 2 }{\frac{\cos x}{e^{\sin^2 x + 1}}}$
14	$y = 2 \cos 3x - \frac{\ln \cos x }{\ln(1 + x^2)} + e^{\cos^2 x \sin x}$	29	$y = x - \frac{10 \sin x}{\ln \left(\frac{e^{-2x}}{2 \sin^2 x + 1} \right)} + x^4 - x^5 $
15	$y = e^{\sin^2 x + 1} - \frac{x}{\ln \cos^3 x + 5 } + (1 + \nu)^x$	30	$y = 3^x + 4x - \frac{\ln \left \frac{\cos^2 x + 1}{x^2 + 1} \right }{2 \sin x + 25}$

Задание И2.2. Составить программу для решения задач, согласно варианту.

1. Из круга радиуса r вырезан прямоугольник, большая сторона которого равна a . Найти максимальный радиус круга, который можно вырезать из полученного прямоугольника.
2. Владелец автомобиля приобрел новый карбюратор, который экономит 50% топлива, новую систему зажигания, которая экономит 30% топлива, и поршневые кольца, экономящие 20% топлива. Верно ли, что его автомобиль теперь сможет обходиться совсем без топлива? Найти фактическую экономию для произвольно заданных сэкономленных процентов.

3. В пространстве заданы два вектора своими координатами. Найти угол (в градусах) между векторами A и B .
4. В пространстве заданы два вектора своими координатами. Найти объем пирамиды, построенной на векторах A , B , C .
5. На тело действуют две силы, заданные векторами A и B . Найти величину и направление (углы с координатными осями) их равнодействующей.
6. Животновод в начале каждой зимы повышает отпускную цену на молоко на $p\%$, а каждым летом – снижает на столько же процентов. Изменится ли цена на молоко и если да, то в какую сторону и на сколько через N лет?
7. Пловцу надо под прямым углом к фарватеру преодолеть реку Урал шириной b м. Его скорость в стоячей воде v_1 м/с; скорость течения реки – v_2 м/с. Под каким углом к фарватеру он должен плыть, чтобы его «не снесло»? Сколько времени займет переправа? Как изменится решение, если посредине реки пловец устал, и его скорость с v_1 м/с упала до v_3 м/с?
8. Сколько кругов заданного радиуса r можно вырезать из правильного треугольника со стороной a ?
9. Какова должна быть длина стороны правильного треугольника a , чтобы из него можно было вырезать N кругов радиуса r ?
10. Студент съедает торт за пять минут, преподаватель – за полчаса, а заведующий кафедрой – за час. За какое время они съедят торт вместе?
11. Правительство гарантирует, что процент инфляции в новом году составит – $p\%$ в месяц. Какого роста цен за год можно ожидать?
12. Спортсмен в первый день пробежал 10 км. Каждый следующий день он увеличивал дневную норму на 10% от нормы предыдущего дня. Какой путь пробежит спортсмен в 7 день? Определить, через сколько дней спортсмен будет пробегать более 20 км? Определить, через сколько дней спортсмен пробежит суммарный путь более 100 км?
13. В бассейн поступает холодная и горячая вода. Холодной водой бассейн заполняется за 6 минут 40 секунд, а горячей водой – за 8 минут. Вся вода вытекает из полного бассейна за 13 минут 20 секунд. Сколько времени понадобится, чтобы наполнить бассейн полностью, при условии, что открыты оба крана, но слив открыт?
14. Студенты варили варенье, но забыли закрыть окно. С улицы в открытое окно залетали пчелы. В первый час на варенье прилетела 1 пчела, во второй час – 2 пчелы, а в третий – 3 пчелы и т.д. Сколько пчел залетело на варенье за сутки, если окно не закрывали?
15. Старинная задача. Летела стая гусей, а навстречу им летит один гусь и говорит: Здравствуй, сто гусей!» А передний старый гусь ему и отвечает: «Нет, нас не сто гусей! Вот если б нас было еще столько, да

- еще полстолька, да еще четверть столько, да ты, гусь, то было бы сто гусей, а теперь ... Вот и рассчитай-ка, сколько нас?»
16. Старинная народная задача. Крестьянка пришла на базар продавать яйца. Первая покупательница купила у нее половину всех яиц и еще пол-яйца. Вторая покупательница приобрела половину оставшихся яиц и еще пол-яйца. Третья купила всего одно яйцо. После этого у крестьянки не осталось ничего. Сколько яиц она принесла на базар?
 17. Муж сообщил своей жене, что отныне каждый месяц им следует экономить вдвое больше, чем в предыдущем месяце. Они экономят 1 доллар в первый месяц, 2 доллара во второй месяц, 4 доллара в третий месяц и так далее. Сколько они сэкономят за год?
 18. Если бы могли обойти земной шар по экватору, то макушка нашей головы описала бы более длинный путь, чем наши ноги. Как велика эта разница?
 19. В 5 мисках – 100 орехов. В первой и второй мисках вместе 52 ореха. Во второй и в третьей – 43, в третьей и четвертой – 34, в четвертой и пятой – 30. Сколько орехов в каждой миске?
 20. Лошадиный барышник на ярмарке купил лошадь за m рублей, а продал за n . Через некоторое время он купил ту же лошадь за h рублей, а продал за q . Каков его барыш?
 21. Скорый поезд вышел из Москвы в Санкт-Петербург и шел без остановок со скоростью v , км/ч. Другой поезд вышел ему навстречу из Санкт-Петербурга и тоже шел без остановок, но со скоростью g , км/ч. На каком расстоянии друг от друга будут поезда за t часов до встречи?
 22. У n хозяев по n кошек, каждая кошка съедает по n мышей, каждая мышь съедает по n колосьев ячменя, из каждого колоса может вырасти по n мер зерна. Сколько мер зерна сохраняется благодаря этим кошкам?
 23. Между городами А и В S километров. Из них выехали два велосипедиста и со скоростью v , км/ч каждый помчались навстречу друг другу. Вместе с первым велосипедистом из города А стартовала муха со скоростью g , км/ч. Встретившись с велосипедистом из города В, муха развернулась и полетела к первому, встретившись с ним, опять полетела ко второму. Когда велосипедисты съехались и остановились, муха утомилась и села одному из них на голову. Сколько километров пролетела муха?
 24. Пасли ребята коней. Если пересчитать ноги коней и детей, получится n , а если головы, то m . Сколько было ребят и сколько коней?
 25. В жаркий день n косцов выпили бочонок кваса за m часов. Сколько косцов за t часа выпьют такой же бочонок кваса?
 26. Вол съел копну за t , конь – за m , коза – за n часов. За сколько времени вол, конь и коза – съедят ту копну вместе?

27. Веселый француз пришел в трактир с неизвестною суммой своего богатства, занял у хозяина столько денег, сколько у себя имел; из сей суммы издержал n рублей. С остатком пришел в другой трактир, где опять, занявши столько, сколько имел, издержал в оном также n рубль; то же учинил в третьем и четвертом трактирах. По выходе из четвертого не имел уже ничего. С какой суммой пришел он в первый трактир?
28. Некто за n алтын купил сукна три четверти аршина. Сколько алтын стоят m аршин? (400 алтын = 1200 коп. = 12 руб.).
29. В такси одновременно сели три пассажира. Когда вышел первый пассажир, на счетчике было $p1$ рублей; когда вышел второй – $p2$ рублей. Сколько должен был заплатить каждый пассажир, если по окончании поездки счетчик показал $p3$ рублей? Плата за посадку составляет $p0$ рублей.
30. Коммерсант, имея стартовый капитал k рублей, занялся торговлей, которая ежемесячно увеличивает капитал на p %. Через сколько лет он накопит сумму s , достаточную для покупки собственного магазина?
31. Селекционер вывел новый сорт зерновой культуры и снял с опытной делянки k кг семян. Посеяв 1 кг семян, можно за сезон собрать p кг семян. Через сколько лет селекционер сможет засеять новой культурой поле площадью s га, если норма высева n кг/га?
32. За первый год производительность труда на предприятии возросла на $p1$ %, за второй и третий – соответственно на $p2$ и $p3$ %. Найти среднегодовой прирост производительности (в процентах).
33. Заданы уравнения двух пересекающихся прямых на плоскости: $y=k1 \cdot x+b1$ и $y=k2 \cdot x+b2$. Найти (в градусах и минутах) угол между ними.

Лабораторная работа № 3

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ВЕТВЛЕНИЯ

В линейной программе все операторы выполняются последовательно, один за другим. Для того чтобы в зависимости от исходных данных обеспечить выполнение разных последовательностей операторов, применяются операторы ветвления *if* и *switch*. Оператор *if* обеспечивает передачу управления на одну из двух ветвей вычислений, оператор *switch* – на одну из произвольного числа ветвей.

Условный оператор if

Используется для разветвления процесса вычислений на два направления. Структурная схема оператора приведена на рис. 3.1.

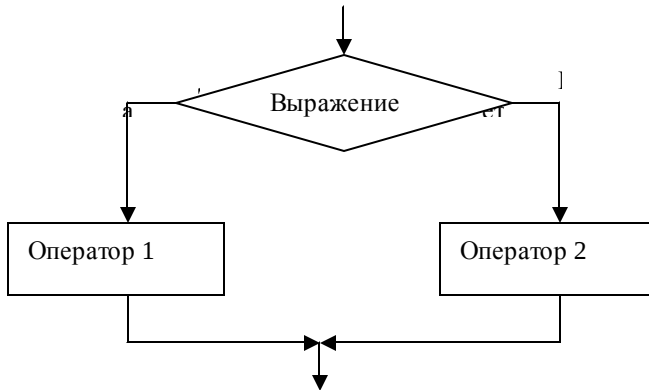


Рис. 3.1

Формат оператора:

if (выражение) оператор_1; [else оператор_2;]

Сначала вычисляется выражение, которое может иметь арифметический тип или тип указателя. Если оно принимает значение не равное нулю, выполняется первый оператор, иначе второй. После этого управление передается на оператор, следующий за условным.

Одна из ветвей может отсутствовать (опускается ключевое слово *else*). Если требуется выполнить несколько операторов, их необходимо заключить в блок ({}), иначе компилятор не сможет понять, где заканчивается ветвление. Блок может содержать любые операторы, в том числе и описания и другие условные операторы (но не может состоять из одних описаний). Необходимо учитывать, что переменная, описанная в блоке, вне блока не существует.

Задача 3.1. Расчет значений кусочных функций

Определить значение кусочной функции для любого заданного x .

$$y = \begin{cases} x-5, & x \leq 0, \\ \frac{\ln x}{x^2 + x}, & 0 < x < 3, \\ \sqrt{x+5}, & x \geq 3. \end{cases}$$

Выбор переменных программы: аргумент и значение функции могут принимать любые значения (2; 2,5; 3,75), все переменные определяем как действительные числа.

Схема алгоритма

Составим блок-схему этой задачи (рис. 3.2).

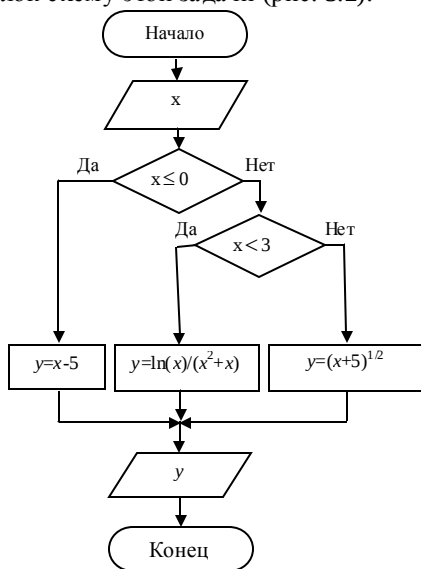


Рис. 3.2

Листинг программы

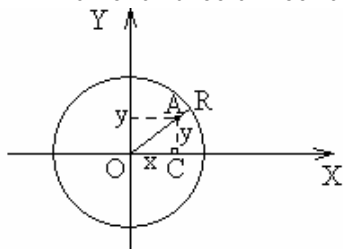
```

#include <stdio.h>
#include <math.h>
int main()
{
    float x,y; // Аргумент и значение функции, переменные вещественного
    типа
    printf("Enter x ");
    scanf("%f",&x); // Ввод переменной x
    if (x<=0)
        y=x-5;
    else if (x<3)
        y=log(pow(x,2)+x);
    else
        y=pow(x+5,1/2.);
}
  
```

```
printf("y(%2.1f)=%2.1f",x,y); // Вывод результата в виде y(x)=y
getchar();getchar();
return 0;
}
```

Задача 3.2. Написать программу, которая определяет по введенным координатам, попадает ли заданная точка в окружность, с центром в точке О (0,0) и заданным радиусом R.

Математическая постановка задачи



Заданная точка имеет координаты – x, y. Рассмотрим треугольник АОС. $\angle OCA = 90^\circ$, $OC = x$, $AC = y$, следовательно, по теореме Пифагора AO^2 (гипотенуза) $= OC^2 + AC^2$. $AO^2 = x^2 + y^2$. Поэтому условие принадлежности точки окружности можно записать в виде:

$$x^2 + y^2 \leq R^2$$

Выбор переменных программы

Из приведенного выше решения определяем следующие переменные: исходные данные – радиус окружности (R) и координаты точки (x и y); результат – сообщение – «находится точка в окружности или нет». Так как радиус окружности и координаты точки могут принимать любые значения (2; 2,5; 3,75), все переменные определяем как действительные числа.

Схема алгоритма

Составим блок-схему этой задачи (рис. 3.3).

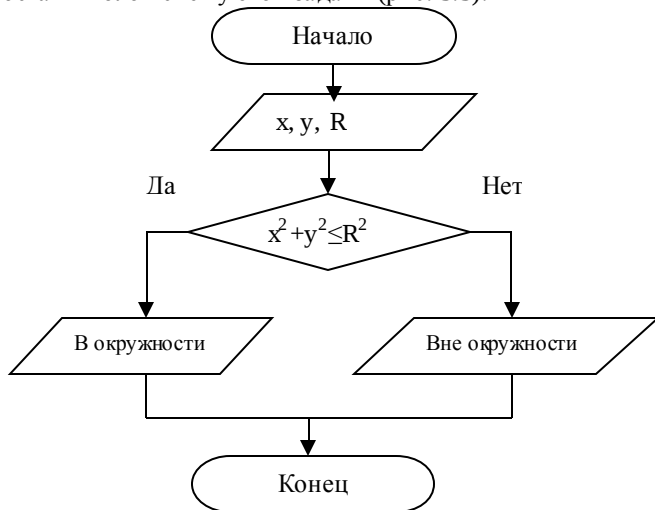


Рис. 3.3

Листинг программы

```

#include <stdio.h>
int main()
{
    float x,y,R; //Координаты точки и радиус, переменные
    вещественного типа
    printf("Введите X, Y, R ->");
    scanf("%f%f%f",&x,&y, &R); //Ввод переменных x,y,R
    if (x*x+y*y<=R*R)
        printf("A(%2.1f,%2.1f) принадлежит окружности (R=%2.1f)",
x,y,R);
    else
        printf("A(%2.1f,%2.1f) не принадлежит окружности (R=%2.1f)",
x,y,R);
    getchar();getchar();
    return 0;
}

```

Тесты

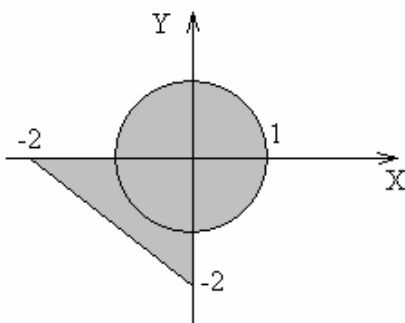
Для испытания программы можно взять тесты, представленные в табл. 3.1.

Таблица 3.1

Тесты для задачи 3.2

x	y	R	Результат
0	0	0	Принадлежит
2	2	1	Не принадлежит
1	1	2	Принадлежит

Задача 3.3. Написать программу, которая по введенным координатам точки определяет, попадает ли эта точка в заштрихованную область.



Математическая постановка задачи

Запишем условия попадания точки в область в виде формул. Область можно описать как круг, пересекающийся с треугольником. Точка может попадать либо в круг, либо в треугольник, либо в их общую часть:

$$\{x^2 + y^2 \leq 1\} \text{ или } \begin{cases} x \leq 0 \\ y \leq 0 \\ y \geq -x - 2 \end{cases}$$

Первое условие задает попадание точки в круг, второе – в треугольник.

Выбор переменных программы

Из приведенного выше решения определяем следующие переменные:

исходные данные – координаты точки (x и y);

результат – сообщение – «находится точка в заштрихованной области?».

Так как координаты точки могут принимать любые значения (2; 2,5; 3,75), все переменные определяем как действительные числа.

Схема алгоритма

Составим блок-схему этой задачи (рис. 3.4).

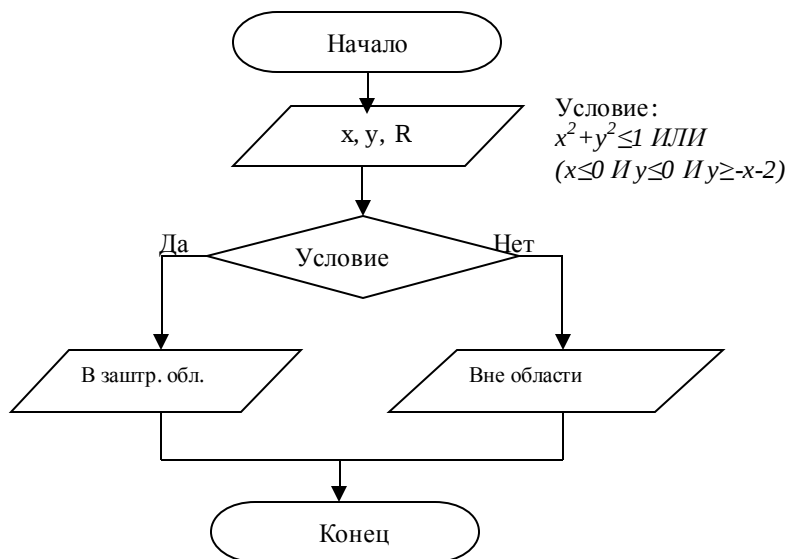


Рис. 3.4

Листинг программы

```

#include <stdio.h>
main()
{
    float x,y; //Координаты точки
    printf("Введите X, Y ->");
    scanf("%f%f",&x,&y);
    if (x*x+y*y<=1 || x<=0 && y<=0 && y>=-x-2)

```

```

    printf("A(%2.1f,%2.1f) принадлежит ", x,y);
else
    printf("A(%2.1f,%2.1f) не принадлежит ", x,y);
getchar();getchar();
return 0;
}

```

Тесты

Для испытания программы можно взять тесты, представленные в табл.3.2.

Таблица 3.2

Тесты для задачи 3.3

<i>x</i>	<i>y</i>	<i>Результат</i>
0	0	Принадлежит
1	1	Не принадлежит
-1	-1	Принадлежит

Оператор выбора switch

Оператор *switch* (переключатель) предназначен для разветвления процесса вычислений на несколько направлений. Формат оператора:

```

switch (выражение)
{
case    константное_выражение_1: [список_операторов_1]
case    константное_выражение_2: [список_операторов_2]
...
case    константное_выражение_n: [список_операторов_n]
[default: операторы]
}

```

Выполнение оператора начинается с вычисления выражения (оно должно быть целочисленным), а затем управление передается первому оператору из списка, помеченного константным выражением, значение которого совпало с вычисленным. После этого, если выход из переключателя явно не указан, последовательно выполняются все остальные ветви.

Выход из переключателя обычно выполняется с помощью оператора *break*.

Все константные выражения должны иметь различные значения, но быть одного и того же целочисленного типа.

Задача 3.4. Составить программу, реализующую действия простейшего калькулятора.

Выбор переменных программы

Определяем следующие переменные:

исходные данные – операнды (*a* и *b*) и знак операции (*op*);
результат (*res*).

Операнды определяем как действительные числа, а знак операций как символ.

Листинг программы

```
#include <stdio.h>
main()
{
    float a,b; //Операнды
    float res; //Результат
    char op; //Знак операции
    bool f=true;
    printf("Input op ->");
    scanf("%c",&op);
    printf("Input a, b ->");
    scanf("%f%f",&a,&b);
    switch (op)
    {
        case '+':
            res=a+b;
            break;
        case '-':
            res=a-b;
            break;
        case '*':
            res=a*b;
            break;
        case '/':
            res=a/b;
            break;
        default :
            printf("No %c operation", op);
            f=false;
    }
    if (f)
        printf("%2.1f %C %2.1f = %2.1f", a,op,b,res);
    getchar();getchar();
    return 0;
}
```

Индивидуальные задания

по теме «Программирование алгоритмов условного перехода»

Задание ИЗ.1. Расчет значений кусочных функций.

Определить значение кусочной функции для любого заданного x .

$$1. \quad y = \begin{cases} x^2, & x \leq 0, \\ \cos x, & 0 < x < 5, \\ \sqrt{x}, & x \geq 5. \end{cases}$$

$$2. \quad y = \begin{cases} x^2 - 4, & x \leq 0, \\ \cos x + 5, & 0 < x < 5, \\ \sqrt{x} + x^2, & x \geq 5. \end{cases}$$

$$3. \quad y = \begin{cases} e^x + x^2, & x \leq 0, \\ \cos x, & 0 < x < 5, \\ 3x - \sqrt{x}, & x \geq 5. \end{cases}$$

$$4. \quad y = \begin{cases} \ln|x|, & x < 0, \\ \cos x, & 0 \leq x < 3, \\ x^{-5/6}, & x \geq 3. \end{cases}$$

$$5. \quad y = \begin{cases} e^{-x}, & x \leq 0, \\ \sin 2x, & 0 < x < 5, \\ \sqrt{x}, & x \geq 5. \end{cases}$$

$$6. \quad y = \begin{cases} \sin x + x^2, & x \leq 0, \\ \cos x, & 0 < x < 5, \\ e^x + \sqrt{x}, & x \geq 5. \end{cases}$$

$$7. \quad y = \begin{cases} \sqrt[3]{x^2}, & x < 0, \\ \cos x, & 0 \leq x < 5, \\ x + 3, & x \geq 5. \end{cases}$$

$$8. \quad y = \begin{cases} e^{-2x}, & x \leq 0, \\ \cos(x - \frac{\pi}{6}), & 0 < x < 5, \\ \sqrt{x}, & x \geq 5. \end{cases}$$

$$9. \quad y = \begin{cases} x^4, & x \leq \pi, \\ \cos x, & \pi < x < 5, \\ \sqrt{x}, & x \geq 5. \end{cases}$$

$$10. \quad y = \begin{cases} e^x, & x \leq -5, \\ 0, & -5 < x < 5, \\ e^{-x+1}, & x \geq 5. \end{cases}$$

$$11. \quad y = \begin{cases} x^2, & x \leq -\frac{3}{2}, \\ \cos(x), & -\frac{3}{2} < x < 5, \\ \sqrt{|\sin x|}, & x \geq 5. \end{cases}$$

$$12. \quad y = \begin{cases} x^2, & x \leq 6, \\ e^{-x}, & 6 < x < 25, \\ 4, & x \geq 25. \end{cases}$$

$$13. \quad y = \begin{cases} x - 5, & x \leq 0, \\ 2\cos x, & 0 < x < 10, \\ 5 + \sqrt{x}, & x \geq 10. \end{cases}$$

$$14. \quad y = \begin{cases} \sin 2x, & x \leq -2\pi, \\ e^x + e^{-x}, & -2\pi < x < 5, \\ \cos 2x, & x \geq 5. \end{cases}$$

$$15. \quad y = \begin{cases} x^2, & x \leq 0, \\ \ln x, & 0 < x < 5, \\ \sqrt{x} - 5x, & x \geq 5. \end{cases}$$

$$16. \quad y = \begin{cases} x + 5, & x < 0, \\ e^x, & 0 \leq x \leq 1, \\ \sin x, & x > 1. \end{cases}$$

$$17. \quad y = \begin{cases} 3x^2, & x \leq -1, \\ 5x, & -1 < x \leq 3, \\ x - 4, & x > 3. \end{cases}$$

$$18. \quad y = \begin{cases} \sin x, & x < 0, \\ 10, & x = 0, \\ \cos x, & x > 0. \end{cases}$$

$$19. \quad y = \begin{cases} x^2 + 22, & x < -5, \\ e^x, & -5 \leq x \leq 5, \\ \ln x, & x > 5. \end{cases}$$

$$20. \quad y = \begin{cases} x^2 - 5, & x \leq -2, \\ \cos 2x, & -2 < x < 1, \\ 15 + \sqrt{x}, & x \geq 1. \end{cases}$$

$$21. \quad y = \begin{cases} \sin(x - 5), & x \leq 0, \\ \frac{\cos 2x}{x^2 + 1}, & 0 < x < 3, \\ \sin(5 + \sqrt{x}), & x \geq 3. \end{cases}$$

22.

$$y = \begin{cases} x^2 - 3x + 9, & x \leq 3, \\ \frac{1}{x^3 + 6}, & 3 < x < 6, \\ \sqrt{x}, & x \geq 6. \end{cases}$$

23.

$$y = \begin{cases} -x^2 - 3x + 9, & x \geq 3, \\ \frac{1}{x^3 - 6}, & 3 > x \geq 2, \\ \sqrt{|x|}, & x < 2. \end{cases}$$

24.

$$y = \begin{cases} \frac{x-5}{\cos^2 x + 1}, & x \leq -5, \\ \ln(x^2 + 1), & -5 < x < 5, \\ 5 + \sqrt{x^2 + x + 10}, & x \geq 5. \end{cases}$$

25.

$$y = \begin{cases} \sin(x-5), & x \leq 0, \\ 2 \cos|x + x^2|, & 0 < x < 1, \\ (5 + \sqrt{x}) \sin x^2, & x \geq 1. \end{cases}$$

26.

$$y = \begin{cases} x - 3, & x \leq 3, \\ \ln x + 2, & 3 < x < 6, \\ \sqrt[3]{x}, & x \geq 6. \end{cases}$$

27.

$$y = \begin{cases} (x-5)^3, & x \leq 2, \\ \sqrt[4]{x^3 - x^2 - 4}, & 2 < x < 8, \\ 5 + \sqrt{x^2 + x + 4}, & x \geq 8. \end{cases}$$

28.

$$y = \begin{cases} \sqrt{|x-5|}, & x \leq 0, \\ \sqrt{2 \cos x}, & 0 < x < 1, \\ \sqrt{x}, & x \geq 1. \end{cases}$$

29.

$$y = \begin{cases} x - 5, & x \leq 0, \\ \frac{\ln x}{x^2 + x}, & 0 < x < 3, \\ \sqrt{x+5}, & x \geq 3. \end{cases}$$

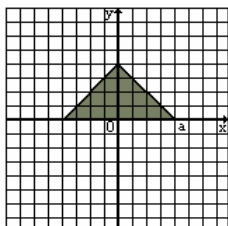
30.

$$y = \begin{cases} \frac{\sqrt{|x-5|}}{x^2 + 1}, & x \leq 0, \\ \frac{x+3}{5 + \sqrt{x}}, & 0 < x < 4, \\ \frac{5 + \sqrt{x}}{\sqrt{x^4 + x^3 + x + 5}}, & x \geq 4. \end{cases}$$

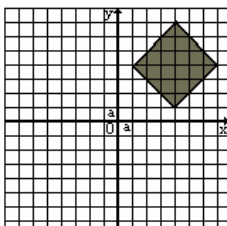
31.

$$y = \begin{cases} \frac{e^x}{x^2 + 1}, & x \leq -3, \\ (x+3)^2, & -3 < x < 5, \\ \frac{5 + \ln(\sqrt{x})}{\sqrt{x^4 + x + 5}}, & x \geq 5. \end{cases}$$

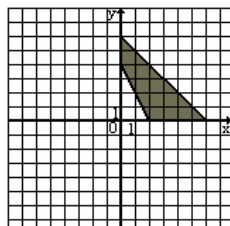
Задание И2.2. Программирование геометрического места точек. Составить программу для определения принадлежности точки заданной области.



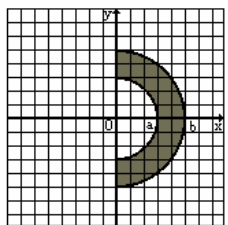
1.



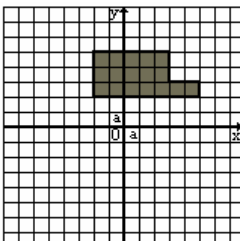
5.



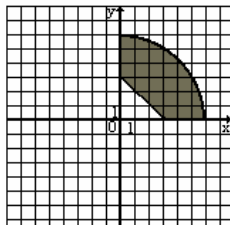
9.



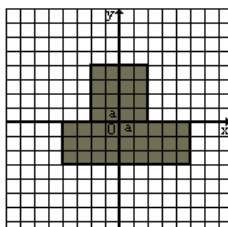
2.



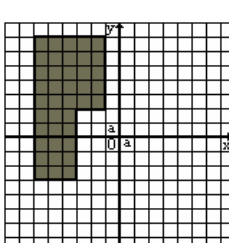
6.



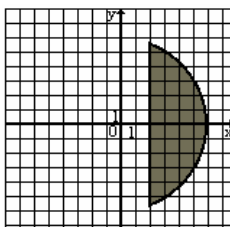
10.



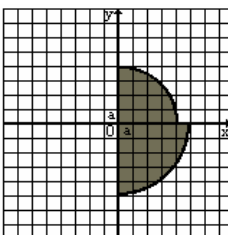
3.



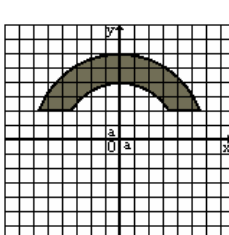
7.



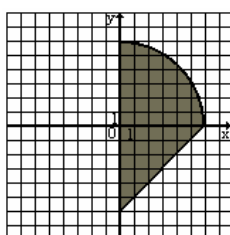
11.



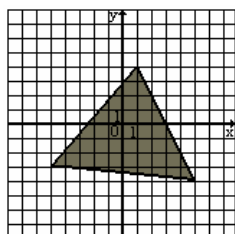
4.



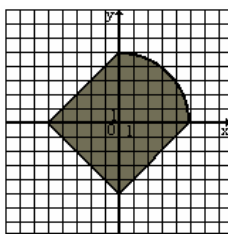
8.



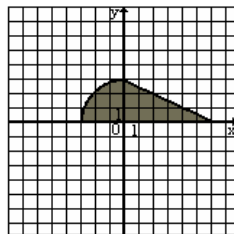
12.



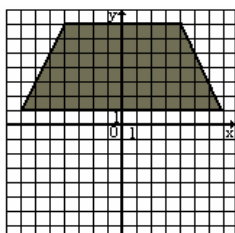
13.



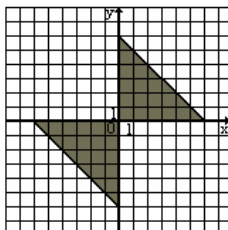
17.



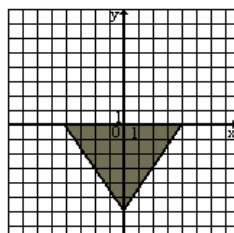
21.



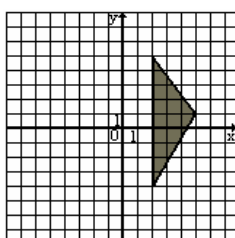
14.



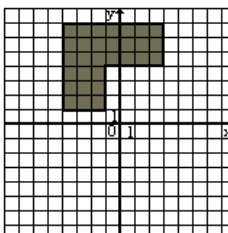
18.



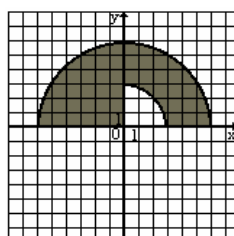
22.



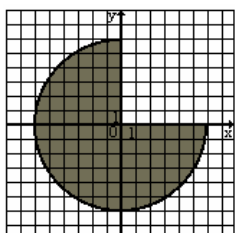
15.



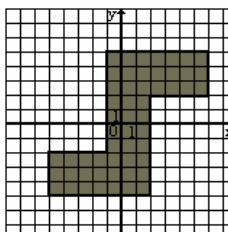
19.



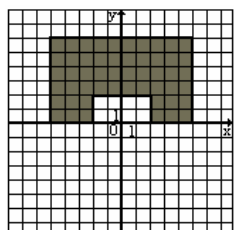
23.



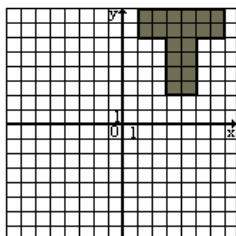
16.



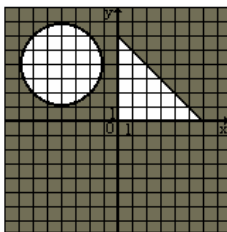
20.



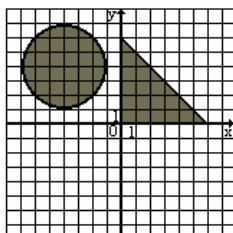
24.



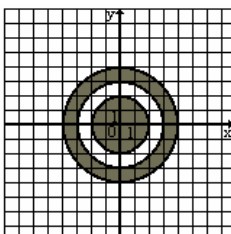
25.



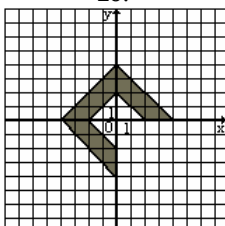
29.



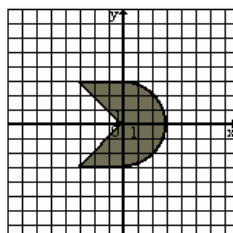
26.



30.



27.



28.

Задание ИЗ.3.

1. Задан четырехугольник со сторонами A, B, C, D . Определить, является ли он параллелограммом или ромбом.

2. На плоскости задан отрезок координатами своих концов. Определить, имеет ли он общие точки с осями координат.

3. На плоскости задан отрезок координатами своих концов. Определить, принадлежит ли отрезок полностью первой четверти координатной плоскости.

4. На плоскости задан отрезок координатами своих концов. Определить, параллелен ли он осям координат.

5. На плоскости задана окружность с центром в точке (X_1, Y_1) , радиусом R . Определить, принадлежит ли точка с координатами X, Y заданному кругу, лежит на ее границе или вне ее.

6. На плоскости задана точка с координатами X, Y . Определить, какой четверти принадлежит точка.

7. Задан координатами своих концов отрезок на плоскости. Определить, имеет ли он длину больше L_1 и меньше L_2 ($L_1 < L_2$).

8. На плоскости заданы две концентрические окружности радиусами, R_2 , с центром в точке (X, Y) . Определить, принадлежит ли точка X_1, Y_1 кольцу между двумя окружностями.

9. Даны три отрезка A, B, C . Определить, существует ли треугольник со сторонами, равными данным отрезкам.

10. Даны три отрезка A, B, C . Определить, является ли треугольник, со сторонами, равным отрезкам, остроугольным, прямоугольным или тупоугольным.

11. Задан треугольник координатами своих вершин. Определить, является он остроугольным, прямоугольным или тупоугольным.

12. Задан треугольник координатами своих вершин. Определить, является ли он тупоугольным с периметром больше заданного числа M .

13. Задана точка на плоскости и два прямоугольника со сторонами, параллельными осям координат. Определить, принадлежит точка только первому прямоугольнику, только второму или обоим вместе.

14. На плоскости заданы точка, круг, радиусом R с центром в начале координат и вписанный в него квадрат со сторонами, параллельными осям координат. Определить, находится точка одновременно и в круге, и в квадрате, вне их или только в круге.

15. На плоскости заданы точка, круг радиусом R с центром в начале координат и описанный вокруг него квадрат со сторонами, параллельными осям координат. Определить, находится точка одновременно и в круге, и в квадрате, вне их или только в круге.

16. На плоскости задана точка, не находящаяся на осях координат. Если точка лежит в круге с центром в начале координат и радиусом R , определить, какой четверти координатной плоскости принадлежит точка.

17. Даны две точки $A(x_1, y_1)$ и $B(x_2, y_2)$. Составить программу, определяющую, которая из точек находится ближе к началу координат.

18. Даны меры двух углов треугольника (в градусах). Определить, существует ли треугольник с такими углами, и если да, то будет ли он прямоугольным.

19. С клавиатуры вводятся длины отрезков a , b , c и d . Оценить их на возможность построения треугольников. В качестве теста взять числа 3, 5, 9, 10.

20. Даны три точки $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$. Определить будут ли они расположены на одной прямой. Если нет, то вычислить $\angle ABC$.

21. На плоскости расположены три точки a , b , c . Определить, какая из точек b или c расположена ближе к a .

22. Даны три положительных числа a , b , c . Проверить, могут ли они быть длинами сторон треугольника. Если да, то вычислить площадь этого треугольника.

23. Дана окружность радиусом R . Определить, впишется ли правильный треугольник со стороной a в эту окружность.

24. Заданы размеры A , B прямоугольного отверстия и размеры x , y , z кирпича. Определить, пройдет ли кирпич через это отверстие, параллельно.

25. Заданы отрезки a , b , c , d . Проверить могут ли они служить сторонами прямоугольника. Если да, то вычислить площадь этого прямоугольника.

26. Дана точка $A(x, y)$. Определить, принадлежит ли она треугольнику с вершинами в точках (x_1, y_1) , (x_2, y_2) , (x_3, y_3) .

27. Два прямоугольника, заданы координатами своих левого верхнего и правого нижнего углов (стороны параллельны осям). Для первого прямоугольника это точки (x_1, y_1) и $(x_2, 0)$, для второго – (x_3, y_3) и $(x_4, 0)$. Составить программу, определяющую, пересекаются ли данные прямоугольники, и вычисляющую площадь общей части, если она существует.

28. На плоскости задана своими координатами точка A . Укажите, где она расположена (на какой оси или в каком координатном угле).

29. Заданы три стороны треугольника a , b и c . Определить, является ли этот треугольник прямоугольным, и какая сторона служит гипотенузой.

30. Найти координаты точек пересечения прямой $y=kx+b$ и окружности радиуса R с центром в начале координат. В каких координатных четвертях находятся точки пересечения? Если точек пересечения нет или прямая касается окружности, выдать соответствующее сообщение.

31. На плоскости задана окружность радиусом R с центром в начале координат и дана точка с координатами X , Y . Определить, принадлежит ли точка кругу, лежит на границе или вне ее.

Задание ИЗ.4. Задачи по теме «Оператор выбора».

1. Составьте программу вычисления функции, если x – натуральное число; a, b, c – действительные числа:

$$y = \begin{cases} a + bx + cx^2, & x = 1; \\ a \cdot \sin(xb^2), & x = 2; \\ \sqrt{|a + bx|}, & x = 3; \\ a \cdot \ln|bx + cx^2|, & x = 4; \\ e^{-ax} + \sin bx + cx, & x = 5. \end{cases}$$

В противном случае (если x не входит в заданный диапазон) вывести сообщение.

2. Пусть элементами прямоугольного равнобедренного треугольника являются: катет a , гипотенуза b , высота h , опущенная из вершины прямого угла на гипотенузу, площадь S . Составить программу, которая по заданному номеру и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.

3. Написать программу, которая бы по введенному номеру времени года (1 – зима, 2 – весна, 3 – лето, 4 – осень) выдавала соответствующие этому времени года месяцы, количество дней в каждом из месяцев.

4. В старояпонском календаре был принят 12-летний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Написать программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю. (Справка: 1996 г. – год Крысы – начало очередного цикла.)

5. Для целого числа k от 1 до 99 напечатать фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.

6. Написать программу, которая бы по введенному номеру единицы измерения (1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр) и длине отрезка L выдавала бы соответствующее значение длины отрезка в метрах.

7. Написать программу, которая по введенному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные выходные дни (например, если введено число 1, то должно получиться 1 января – Новый год, 7 января – Рождество).

8. Дано натуральное число N . Если оно делится на 4, вывести на экран ответ $N=4k$ (где k – соответствующее частное); если остаток от деления на 4 равен 1 – $N=4k+1$; если остаток от деления на 4 равен 2 –

$N=4k+2$; если остаток от деления на 4 равен 3 – $N=4k+3$. Например, $12=4\cdot3$, $22=4\cdot5+2$.

9. Составить программу, которая на ввод знака препинания выдает на экран дисплея его название. Например, на ввод точки выдает текст: «Точка».

10. Даны два действительных положительных числа x и y . Арифметические действия над числами пронумерованы (1 – сложение, 2 – вычитание, 3 – умножение, 4 – деление). Составить программу, которая по введенному номеру выполняет то или иное действие над числами.

11. Написать программу, которая бы по введенному номеру единицы измерения (1 – килограмм, 2 – миллиграмм, 3 – грамм, 4 – тонна, 5 – центнер) и массе M выдавала бы соответствующее значение массы в килограммах.

12. Пусть элементами равностороннего треугольника являются: сторона a , площадь S , высота h , радиус вписанной окружности r , радиус описанной окружности R . Составить программу, которая по заданному номеру и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.

13. Составить программу для определения подходящего возраста кандидатуры для вступления в брак, используя следующее соображение: возраст девушки равен половине возраста мужчине плюс 7, возраст мужчины определяется соответственно как удвоенный возраст девушки минус 14.

14. Напишите программу, которая по заданной дате определяет время года. Программа должна проверять корректность введенной даты.

15. Напишите программу, которая читает натуральное число в десятичном представлении, а на выходе выдает это же число в десятичном представлении и на естественном языке.

Например, 7 семь
 204 двести четыре
 52 пятьдесят два

16. Вычислить номер дня в невисокосном году по заданным числу и месяцу.

17. Даны два комплексных числа x и y . Арифметические действия над числами пронумерованы (1 – сложение, 2 – вычитание, 3 – умножение, 4 – деление). Составить программу, которая по введенному номеру выполняет то или иное действие над числами.

18. Написать программу, которая требует ввода времени дня и, в зависимости от введенного значения, желает доброго утра, доброго дня, доброго вечера или спокойной ночи.

19. Дано натуральное число. Определить остаток от деления на 4 и вывести его в текстовом виде (ноль, один, два, три).

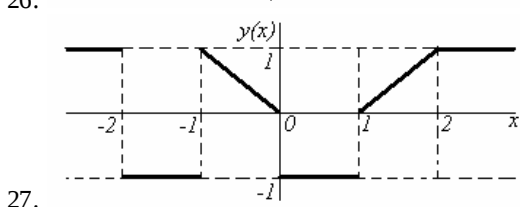
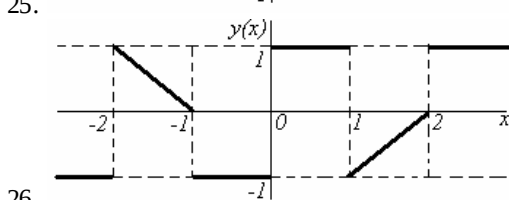
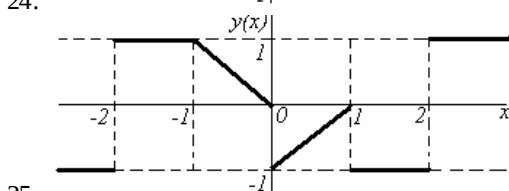
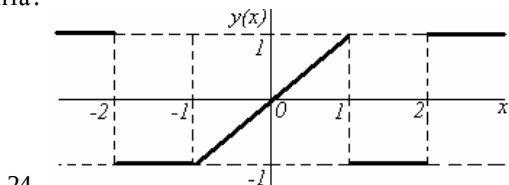
20. Дано неотрицательное число k , не превышающее десяти тысяч. Напечатать фразу « k ворон» русскими словами. (Пример: если $k=23$, то должно быть напечатано «двадцать три вороны»; если $k=3651$, то «три тысячи шестьсот пятьдесят одна ворона»).

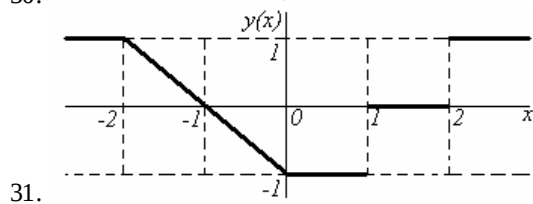
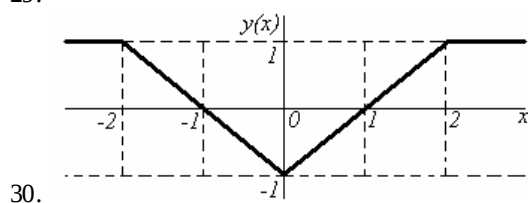
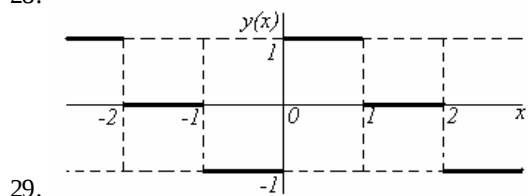
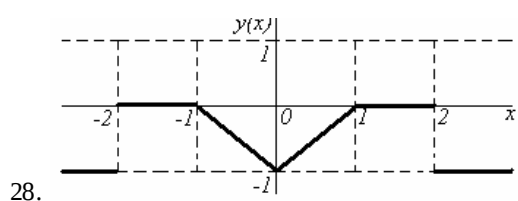
21. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент). Составить программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.

22. Написать программу, которая на ввод знака препинания выдает на экран дисплея его название. Например, на ввод точки выдает текст: «Это точка».

23. Написать программу, которая на название фигуры (треугольник, квадрат, ромб, прямоугольник и т.д.) выводит формулу, по которой вычисляется площадь этой фигуры.

24-31. Записать программу, которая на ввод значения аргумента выдает значение функции, заданной графиком, соответствующим номеру варианта:





Лабораторная работа № 4

ЦИКЛЫ

Операторы цикла используются для организации многократно повторяющихся вычислений. Любой цикл состоит из тела цикла, т.е. тех операторов, которые выполняются несколько раз, начальных установок, модификации параметра цикла и проверки условия продолжения выполнения цикла.

Один проход цикла называется итерацией. Проверка условия выполняется на каждой итерации либо до тела цикла (цикл с предусловием), либо после тела цикла (цикл с постусловием). Тело цикла с предусловием всегда выполняется хотя бы один раз, после чего проверяется необходимость следующего его выполнения. Проверка необходимости выполнения цикла с предусловием делается до тела цикла, поэтому возможно, что он не выполнится ни разу.

Переменные, изменяющиеся в теле цикла и используемые при проверке условия продолжения, называются параметрами цикла. Целочисленные параметры цикла, изменяющиеся с постоянным шагом по каждой итерации, называются счетчиками цикла.

Начальные установки могут явно не присутствовать в программе, их смысл состоит в том, чтобы до входа в цикл задать значения переменным, которые в нем используются.

Цикл завершается, если условие его продолжения не выполняется.

Цикл с предусловием (*while*)

Синтаксис оператора *while* выглядит следующим образом:

while (*выражение*) *оператор*;

Выражение определяет условие повторения тела цикла, представленного простым или составным оператором. Выполнение оператора начинается с вычисления выражения. Если оно истинно (не равно 0), выполняется оператор, после чего управление возвращается заголовку цикла, и все повторяется снова. Когда условие оказывается ложным (равно 0), управление передается следующему после цикла оператору.

Одиночный оператор тела цикла можно заменить блоком, заключенным в фигурные скобки:

```
while (выражение)  
{  
  операторы_тела цикла  
}
```

Распространенный прием программирования – организация бесконечного цикла с заголовком *while* (*true*) либо *while* (*1*) и принудительным выходом из тела цикла по выполнению какого-либо условия.

В круглых скобках после ключевого слова *while* можно вводить описание переменной. Областью ее действия является цикл:

```
while (int x = 0)
{
...
/* область действия x */
}
```

Задача 4.1. Написать программу, которая выводит на экран таблицу умножения на введенное число.

Листинг программы

```
int main()
{
    int n;
    printf("Input n->"); scanf("%d",&n);
    int i=1;
    while (i<=10)
    {
        printf("\n%2d*%2d=%2d",i,n,i*n);
        i++;
    }
    getchar();getchar();
    return 0;
}
```

Цикл с постусловием (do while)

Цикл с постусловием имеет вид:

```
do
оператор
while выражение;
```

Сначала выполняется простой или составной оператор, составляющий тело цикла, а затем вычисляется выражение. Если оно истинно (не равно 0), тело цикла выполняется еще раз. Цикл завершается, когда выражение станет равным 0 или в теле цикла будет выполнен какой-либо оператор передачи управления. Тип выражения должен быть целым или приводимым к нему.

Задача 4.2. Написать программу для решения задачи № 4.1, используя цикл *do...while*.

Листинг программы

```
int main()
{
    int n;
    printf("Input n->"); scanf("%d",&n);
    int i=1;
    do
    {
        printf("\n%2d*%2d=%2d",i,n,i*n);
```

```

    i++;
}
while (i<=10);
getchar();getchar();
return 0;
}

```

Цикл с параметром (for)

Цикл *for* имеет следующий формат:

for (Выражение1; Выражение2; Выражение3) Тело цикла;

Выражение1 обычно используется для установления начального значения переменных, управляющих циклом. *Выражение2* – это выражение, определяющее условие, при котором тело цикла будет выполняться. *Выражение3* определяет изменение переменных, управляющих циклом после каждого выполнения тела цикла.

Схема выполнения оператора *for*:

1. Вычисляется *Выражение1*.
2. Вычисляется *Выражение2*.
3. Если значения *Выражения2* отлично от нуля (истина), выполняется *Тело цикла*, вычисляется *Выражение3* и осуществляется переход к пункту 2, если *Выражение2* равно нулю (ложь), то управление передается на оператор, следующий за оператором *for*.

Существенно то, что проверка условия всегда выполняется в начале цикла. Это значит, что *Тело цикла* может ни разу не выполниться, если условие выполнения сразу будет ложным. Модификации выполняются после каждой итерации цикла и служат обычно для изменения параметров цикла. В части модификаций можно записать несколько операторов через запятую. Простой или составной оператор представляет собой тело цикла. Любая из частей оператора *for* может быть опущена (но точки с запятой надо оставить на своих местах!).

Оператор *for* может быть заменен оператором *while* следующим образом:

```

Выражение1;
while (Выражение2)
{ Тело цикла
  Выражение3;
}

```

Цикл *for* удобен при организации циклов с параметром (переменной, которая при каждом входе в цикл принимает новое значение).

Задача 4.3. Написать программу для решения задачи № 4.1, используя цикл с параметром.

Листинг программы

```

int main()
{

```

```

int n;
printf("Input n->"); scanf("%d",&n);
for (int i=1; i<=10;i++)
    printf("\n%2d*%2d=%2d",i,n,i*n);
getchar();getchar();
return 0;
}

```

Часто встречающиеся ошибки при программировании циклов – использование в теле цикла неинициализированных переменных и неверная запись условия выхода из цикла. Чтобы избежать ошибок, рекомендуется:

- проверить, всем ли переменным, встречающимся в правой части операторов присваивания в теле цикла, присвоены до этого начальные значения (а также возможно ли выполнение других операторов);
- проверить, изменяется ли в цикле хотя бы одна переменная, входящая в условие выхода из цикла;
- предусмотреть аварийный выход из цикла по достижению некоторого количества итераций;
- если в теле цикла требуется выполнить более одного оператора, необходимо заключать их в фигурные скобки.

Индивидуальные задания по теме «Циклы»

Задание И4.1. Цикл с параметром.

1. Дано натуральное число N . Вычислить $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$.

2. Дано натуральное число N . Вычислить:

$$\left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{N^2}\right).$$

3. Дано натуральное число N . Вычислить: $\frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \dots + \frac{2n-1}{2n}$.

4. Даны действительное число a , натуральное число N . Вычислить: $a(a+1)(a+2)(a+3)\dots(a+N-1)$.

5. Даны действительное число a , натуральное число N . Вычислить:

$$\frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1)\dots(a+n)}.$$

6. Даны действительное число a , натуральное число N . Вычислить:

$$\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2n}}.$$

7. Даны действительное число a , натуральное число N . Вычислить: $a(a-N)(a-2N)\dots(a-N^2)$.

8. Вычислить: $(1+\sin 1!)(1+\sin 2!)\dots(1+\sin 10!)$.

9. Для натурального N найти:

$\sin x + \sin^{2!} x + \dots + \sin^{N!} x$, где x – любое число.

10. Для натурального N найти:

$\sin x + \sin x^{2!} + \dots + \sin x^{N!}$, где x – любое число.

11. Для натурального N найти: $1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{N!}$.

12. Для натурального N найти: $\sin \frac{1}{N} + \sin \frac{1}{2N} + \dots + \sin \frac{1}{N^2}$.

13. Для натурального N найти: $\frac{1!}{\cos N} + \frac{2!}{\cos N} + \dots + \frac{N!}{\cos N}$.

14. Для натурального N найти:

$(x^2 + 1)\cos x + (x^2 + 1)\cos 2x + \dots + (x^2 + 1)\cos Nx$, где x – любое число.

15. Для натурального N найти: $\frac{a}{(1+1)!} + \frac{a}{(2+1)!} + \dots + \frac{a}{(N+1)!}$,

где a – любое число.

16. Вычислить $P = \left(1 - \frac{1}{2^2}\right)\left(1 - \frac{1}{3^2}\right)\dots\left(1 - \frac{1}{H^2}\right)$, $H > 2$.

17. Вычислить $y = \cos(1 + \cos(2 + \dots + \cos(39 + \cos(40))\dots))$.

18. Вычислить: $y = \cos x + \cos x^2 + \cos x^3 + \dots + \cos x^n$.

19. Вычислить y , для $n > 1$: $y = 1! + 2! + 3! + \dots + n!$.

20. Вычислить $y = \sin 1 + \sin 2! + \sin 3! + \dots + \sin 10!$.

21. Для натурального N найти:

$\frac{1}{\sin 1} + \frac{1}{\sin(1+2)} + \frac{1}{\sin(1+2+3)} + \dots + \frac{1}{\sin(1+2+3+\dots+N)}$.

22. Для натурального N вычислить:

$\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin N}$.

23 – 31. Составить программу для вычисления суммы ряда:

23. $\sum_{n=1}^{10} \frac{\sqrt[4]{n}}{n+1}$.

24. $\sum_{n=1}^{40} \frac{1}{\sqrt[3]{n} + n}$.

25. $\sum_{n=1}^{10} \operatorname{arctg} \frac{1}{n}$.

26. $\sum_{n=1}^{15} n + e^{-2n}$.

27. $\sum_{n=1}^{40} \frac{\sqrt[3]{n+1}}{n}.$
28. $\sum_{n=1}^{25} \cos^2 \frac{\pi}{4 \cdot n}.$
29. $\sum_{n=1}^{10} \frac{1}{\sqrt[3]{n \cdot (n+1) \cdot (n+2)^3}}.$
30. $\sum_{n=1}^{40} \frac{1}{(2 \cdot n - 1) \cdot 2^{2 \cdot n - 1}}.$
31. $\sum_{n=1}^{100} \frac{1}{n \cdot (n+1)}.$
32. $\sum_{n=1}^{20} \frac{e^{-n}}{(2 \cdot n - 1)}.$
33. $\sum_{n=1}^{10} \frac{n-1}{n \cdot (n+1)}.$

Задание И4.2. Цикл с предусловием и с постусловием.

1. Вычислить y – первое из чисел $\sin(x)$, $\sin(\sin(x))$, $\sin(\sin(\sin(x)))$, ..., меньшее по модулю 10^{-4} .

2. Найти первый отрицательный элемент последовательности $\cos(\text{ctg}(h))$, где $h = 1, 2, 3, \dots$.

3. Дано действительное число E ($E > 0$). Необходимо вычислить следующую сумму: $\sum_{n=1}^{\infty} \frac{1}{3^n} \cos 3^{n-1}$. Следует учесть только те слагаемые, в которых множитель $\frac{1}{3^n}$ имеет величину не меньшую, чем E .

4–13. Не используя стандартные функции, вычислить с точностью $E=10^{-4}$:

4. $y = \text{sh}x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots;$
5. $y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots;$
6. $y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \dots;$
7. $y = \text{arctg}x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots;$

$$8. y = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots;$$

$$9. y = \operatorname{arctg}\left(-\frac{1}{x}\right) = -\frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \frac{1}{7x^7} - \dots, \text{ для } x=1, 2, 3, \dots;$$

$$10. y = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} t_n(x);$$

$$11. y = \frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots = \sum_{n=0}^{\infty} t_n(x);$$

$$12. y = \ln x = 2 \left(\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right) = \sum_{n=0}^{\infty} t_n(x),$$

при $x > 0$;

$$13. y = e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots, \text{ при } |x| < \infty;$$

14. Задано положительное число e ($e < 10^{-2}$). Последовательность $a_1, a_2, \dots, a_n$ образуется по следующему закону:

$$a_i = \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{3}\right) \cdot \dots \cdot \left(1 - \frac{1}{i+1}\right). \text{ Найти первый элемент последовательности, для которого выполнено условие } |a_n - a_{n-1}| < e. \text{ Элементы последовательности вывести в один столбец. Задачу решить без применения массива.}$$

15. Составить программу вычисления функции $\operatorname{ch} x$ по приближенной формуле $\operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$ с точностью $E \geq \left| \frac{x^{2n+1}}{(2n+1)!} \right|$.

16. Составить программу для вычисления количества положительных значений функции $y = \sum_{i=1}^{10} \cos(5i^2 + 8x)$. При x , изменяющемся от 13 до 33 с шагом 1.5, используя цикл `do while`.

17. Найти первый отрицательный элемент последовательности $\sin(\operatorname{ctg}(x_i))$, если x_1 – запрашивается, а $x_{i+1} = x_i + 0.3$.

18. Числа Фибоначчи определяются по формулам: $f_0 = 0, f_1 = 1, f_k = f_{k-2} + f_{k-1}$. Найти первое число Фибоначчи, большее m ($m > 0$).

19. Запрашивается число n . Логической переменной t присвоить значение *true*, если в последовательности $\sin(x^k)$, где $k=1, 2, 3, \dots, n$, есть хотя бы одно отрицательное число, иначе *false*.

20. Распечатать значения функции $f(x) = x - \sqrt{e^{-x}} \sin(x+1)$ до первого значения, отличающегося от 0, не более чем на E , если начальное значение $x=-1$ и изменяется с шагом 0.001.

21. Числа Фибоначчи определяются по формулам: $f_0 = 0, f_1 = 1, f_k = f_{k-2} + f_{k-1}$. Вычислить сумму всех чисел Фибоначчи, которые не превосходят M ($M > 0$).

22. Распечатать значения функции, пока не будет достигнуто пересечение графиков функций $2x^3 - x^2 + 2$; $e^{\frac{x}{3}} + e^{-\frac{x}{3}}$, если начальное значение $x=-2$ и изменяется с шагом 0.01.

23. Дана последовательность целых чисел (не менее 100). Элементы последовательности генерируются случайным образом в диапазоне $[-30; 70]$. Вычислить сумму элементов последовательности, предшествующих «особому» элементу, то есть элементу, который больше своего соседа справа. Задача решается без применения массива.

24. Даны положительные действительные числа A, X, E . В последовательности $y_1, y_2, \dots$, образованной по закону $y_0 = A$; $y_i = \frac{y_{i-1} + \frac{X}{y_{i-1}}}{2}$, $i=1, 2, 3, \dots$, найти первый элемент y_n , для которого выполнено неравенство $|y_n^2 - y_{n-1}^2| < E$.

25. Число π вычисляется по формуле Грегори следующим образом: $\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots\right)$, причем, чем больше слагаемых в скобках, тем выше точность вычисления числа π . Определить минимальное количество слагаемых для вычисления π с точностью 0.001.

26. Дана последовательность, состоящая из дробей: $\frac{1}{1}, \frac{4}{2}, \frac{9}{4}, \frac{16}{8} \dots$. Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

27. Дана последовательность, состоящая из дробей: $\frac{1}{1}, \frac{4}{2}, \frac{7}{3}, \frac{10}{4} \dots$. Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

28. Дана последовательность, состоящая из дробей: $\frac{1}{1}, \frac{3}{2}, \frac{5}{3}, \frac{7}{4}, \dots$

Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

29. Дана последовательность, состоящая из дробей: $\frac{1}{2}, \frac{3}{4}, \frac{5}{6}, \frac{7}{8}, \dots$

Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

30. Дана последовательность, состоящая из дробей: $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots$

Какое минимальное количество элементов последовательности нужно сложить, чтобы сумма превысила заданное число $S > 1$?

31. Найти первый отрицательный элемент последовательности $\sin(x^k)$, где $k=1, 2, 3, \dots$

Задание И4.3. Дополнительные задачи.

1. Дано натуральное число n . Найти сумму первой и последней цифры этого числа.

2. Дано натуральное число n . Переставив местами первую и последнюю цифры этого числа, получить новое число.

3. Даны два натуральных числа m и n ($m \leq 9999, n \leq 9999$). Проверить, есть ли в записи числа m цифры, совпадающие с цифрами в записи числа n .

4. Дано натуральное число $n \leq 99$. Получить новое число, дописав к нему цифру k в конец и в начало.

5. Даны натуральные числа n, k . Проверить, есть ли в записи числа n^k (не более 10^9) цифра m .

6. Среди всех n -значных чисел указать те, сумма цифр которых равна данному числу k .

7. Найти наибольшую и наименьшую цифры в записи данного натурального числа.

8. Найти на отрезке $[n, m]$ минимальное натуральное число, имеющее наибольшее количество делителей.

9. Дано целое число в диапазоне от 0 до 1000000. Проверить, присутствуют ли одновременно в записи числа цифры 1 и 5.

10. Дано целое число в диапазоне от 0 до 1000000. Проверить, отсутствуют ли одновременно в записи числа цифры 2 и 6.

11. Дано целое число в диапазоне от 0 до 1000000. Проверить, присутствуют ли в записи числа хотя бы одна цифра кратная трем.

12. Дано целое число в диапазоне от 0 до 1000000. Проверить, отсутствуют ли в записи числа четные цифры.

13. Найти произведение цифр заданного k -значного числа.

14. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести цифры этого числа в порядке неубывания (например, 546085 $\Rightarrow$ 045568). Функции работы со строками не использовать.

15. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести цифры этого числа в порядке невозрастания (например, 546085 $\Rightarrow$ 865540). Функции работы со строками не использовать.

16. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести цифры этого числа в обратном порядке (например, 5485 $\Rightarrow$ 5845). Функции работы со строками не использовать.

17. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести новое число, полученное из K вычеркиванием всех четных цифр (например, 234583 $\Rightarrow$ 353). Функции работы со строками не использовать.

18. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести это число без первой и последней цифры (например, 234583 $\Rightarrow$ 3458). Функции работы со строками не использовать.

19. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти сумму всех четных цифр этого числа. Функции работы со строками не использовать.

20. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти сумму всех нечетных цифр этого числа. Функции работы со строками не использовать.

21. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти произведение всех цифр этого числа, которые кратны трем. Функции работы со строками не использовать.

22. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти сумму всех цифр этого числа, больших заданного T ($0 \leq T \leq 9$). Функции работы со строками не использовать.

23. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти количество нулей в цифровой записи числа. Функции работы со строками не использовать.

24. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти количество цифр в цифровой записи данного числа, которые имеют наименьшее значение (например, 956569 – количество цифр с наименьшим значением равно 2 – две цифры 5).

25. Дано произвольное целое положительное число K ($K \leq 10^9$). Найти количество цифр в K равных T ($0 \leq T \leq 9$). Функции работы со строками не использовать.

26. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести новое число, полученное из K путем замены последней цифры на значение наибольшей цифры (например, 2854353 $\Rightarrow$ 2354358). Функции работы со строками не использовать.

27. Дано произвольное целое положительное число K ($K \leq 10^9$). Вывести новое число, полученное из K вычеркиванием всех единиц (например, 2134513 $\Rightarrow$ 23453). Функции работы со строками не использовать.

Задание И4.4. Сложные задачи.

1. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить минимальное количество билетов, расположенных между двумя «счастливыми» билетами («счастливым» билетом считать билет, сумма первых трех цифр которого равна сумме трех последних цифр).

2. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить максимальное количество билетов, расположенных между двумя «счастливыми» билетами («счастливым» билетом считать билет, сумма первых трех цифр которого равна сумме трех последних цифр).

3. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить количество билетов, расположенных между первым и последним «счастливыми» билетами («счастливым» билетом считать билет, сумма первых трех цифр которого равна сумме трех последних цифр).

4. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить количество «счастливых» билетов («счастливым» билетом считать билет, сумма первых трех цифр которого равна сумме трех последних цифр).

5. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить минимальное количество билетов, расположенных между двумя билетами с номерами – палиндромами (палиндромом является симметричное число, например: 234432 или 254452).

6. Дана лента билетов, содержащих шестirazрядные номера от N до M . Определить максимальное количество билетов, расположенных между двумя билетами с номерами – палиндромами (палиндромом является симметричное число, например: 234432 или 254452).

Задание И4.5. Задачи на моделирование.

1. На заводе собирают прибор из трех блоков. Известно, что среди блоков первого типа встречаются 2% со скрытыми дефектами, среди блоков второго и третьего типа – соответственно 3 и 5% дефектных. С использованием генератора случайных чисел промоделировать сборку 1000 деталей и определить, сколько будет собрано приборов без брака.

2. Шахматист A в среднем на каждые 100 партий выигрывает у шахматиста B на шесть партий больше, чем проигрывает, а доля ничьих равна 80%. С использованием генератора случайных чисел промоделировать матч из 24 партий. С каким результатом он закончится?

3. Составить программу, позволяющую промоделировать опрос 100 человек и на его основании выяснить: переименовать село Папинск в село Маминск или оставить за ним прежнее название. Примечание: программа должна генерировать ответы самостоятельно с использованием генератора случайных чисел.

4. Имеются две спичечные коробки, в каждой из которых находится по десять спичек. Случайным образом выбирается коробка и из нее достается одна спичка. Процесс продолжается до тех пор, пока одна из коробок не опустеет. С использованием генератора случайных чисел промоделировать этот процесс и ответить на вопрос: сколько спичек будет всего сожжено?

5. В детском саду имеется группа детей из 20 человек. Каждому ребенку на утреннике Дед Мороз случайным образом дарит одну из следующих домашних игрушек: зайца, мяч или куклу. С использованием генератора случайных чисел промоделировать этот процесс и ответить на вопрос: сколько игрушек каждого вида было подарено?

6. Робот находится в центре окружности радиусом 3,5 метра и в каждый момент времени делает шаг (длиной 1 м) в случайном направлении: на север, на юг, на восток или на запад. С использованием генератора случайных чисел промоделировать этот процесс и ответить на вопрос: хватит ли 12 шагов, чтобы выйти за пределы окружности?

7. Известно, что в среднем из 100 выстрелов солдат А поражает мишень 75 раз, а солдат В – 80 раз. С использованием генератора случайных чисел промоделировать соревнование между ними, в котором каждому нужно попасть в цель по 10 раз. Кто быстрее поразит все мишени?

8. В среднем из 128 компьютеров в течение месяца на одном выходит из строя дисплей. За тот же период на одной из 67 ЭВМ происходит поломка дисководов и на двух из 53 машин происходит крах системы из-за заражения вирусом. С использованием генератора случайных чисел смоделировать работу дисплейного класса из 13 компьютеров за один месяц и ответить на вопрос: каково общее количество поломок за этот период?

9. Женщина шла на базар продавать яйца. Ее случайно сбил всадник, в результате чего яйца разбились. Всадник предложил оплатить убытки и спросил, сколько у нее было яиц. Женщина сказала, что точного числа не помнит, но когда она брала яйца парами, то оставалось одно яйцо. Одно яйцо оставалось также, когда она брала по 3, 4, 5 и 6 яиц, но когда она брала по 7 штук, то в остатке ничего не было. Какое минимальное число яиц могло быть в корзине.

10. У гусей и кроликов 64 лапы. Сколько может быть кроликов и гусей (вывести все варианты)?

11. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько амёб будет через 3 часа, 6, 9, 12 и 24 часа.

12. Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней? Сколько дней понадобится спортсмену, чтобы суммарный путь был больше 150 км?

13. Сколько можно купить быков, коров и телят, платя за быка 10 руб., за корову 5 руб., а за теленка – 0,5 руб, если на 100 руб. надо купить 100 голов скота? Выведите все варианты.

14. Ежемесячная стипендия студента составляет A руб., а расходы на проживание превышают стипендию и составляют B руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Составьте программу расчета суммы денег, которую необходимо единовременно попросить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

15. Гусеница ползет по куску резины длиной 7 см, стремясь достичь противоположного конца. Ползет она со скоростью 1 см/мин. Кусок резины может растягиваться до любой длины. Через минуту вы вытягиваете резину так, чтобы она удлинилась вдвое (т.е. стала 14 см в длину). Гусеница прочно держится на поверхности и продолжает двигаться, когда вы тянете резину. Она ползет все с той же скоростью. Еще через минуту вы снова вытягиваете резину так, чтобы ее первоначальная длина утраивается (т.е. она становится равной 21 см). Гусеница продолжает ползти, а вы каждую минуту продолжаете тянуть резину, в четвертый раз уже удлинив ее в четыре раза. Доберется ли гусеница когда-нибудь до противоположного конца? Если да, то когда?

16. Некий мужчина отправляется на службу, которая находится на расстоянии 1 км от дома. Дойдя до места работы, он вдруг вспоминает, что забыл поцеловать жену, и поворачивает назад. Пройдя полпути, он меняет решение, посчитав, что правильнее вернуться на работу. Пройдя $1/3$ км по направлению к конторе, он вдруг осознает, что будет настоящим подлецом, если так и не поцелует жену. На этот раз, прежде чем снова изменить мнение он проходит $1/4$ км. Так он продолжает метаться, и после N -го этапа, пройдя $1/N$ км, снова меняет решение. Это продолжается пока расстояние $1/N$ превышает длину шага, после чего человек останавливается. Определить координату точки остановки.

17. Кролики питаются дарами земли, а лисы поедают кроликов. Изменение в численности популяции кроликов и лис описывается дифференциальным уравнением:

$$\frac{\partial R}{\partial t} = A \cdot R - B \cdot R^2 - C \cdot R \cdot F; \quad \frac{\partial F}{\partial t} = -D \cdot F + E \cdot R \cdot F,$$

где R – количество кроликов; F – количество лис; AR – определяет рождаемость кроликов; BR^2 – смертность кроликов естественным путем; CRF – смертность кроликов за счет съедания лисами; DF – естественная смертность лис; ERF – темп рождаемости лис.

Рассмотреть контрольный пример при $A=0,04$, $B=0,00005$, $C=0,002$, $D=0,03$ и $E=0,0002$. Отобразить изменение численности кроликов и лис по месяцам в течение года, выбрав начальное количество кроликов 340 и лис 35.

18. Составить программу для построения траектории мяча, который начинает двигаться из левого верхнего угла экрана. Уравнение движения имеет вид: $\frac{\partial V}{\partial t} = G - k \cdot V$,

где V – функция изменения скорости; k – коэффициент сопротивления воздуха; G – ускорение свободного падения.

Построить траекторию движения мяча. Проследить влияние k и начальной скорости движения.

19. В магазине стоит очередь из N человек. Время обслуживания i -го покупателя t_i – случайная величина, распределенная по закону равномерной плотности в интервале $[2,5; 10,4]$.

Получить $t_1, t_2, \dots, t_N$ – времена пребывания в очереди каждого покупателя. Указать номер того человека, для обслуживания которого потребовалось минимальное время.

20. Пара носков стоит 1,05 долл.; при покупке дюжины пар взимается плата в размере 10,25 долл., а при покупке 12 дюжин пар – 110 долл. Напишите программу для помощи покупателям, которая рассчитывает минимальную стоимость покупаемых партий носков (например, 13 пар должны стоить 10,25 долл. + 1,05 долл. = 11,30 долл.).

21. Мощность радиоактивного излучения пропорциональна концентрации радиоактивного вещества. Период полураспада одного из изотопов углерода составляет 8 дней. В начальный момент времени мощность излучения составляет 2 рентгена/час. Определить, через сколько дней мощность излучения снизится до величины 0,15 рентгена/час.

Изменение концентрации и, следовательно, мощности излучения описывается формулой

$$Q = Q_0 e^{-\lambda t},$$

где Q_0 – начальная мощность; $\lambda = \frac{\ln 2}{P}$; P – период полураспада.

Для решения задачи необходимо изменять t с заданным шагом, вычислять Q и сравнивать его с допустимой величиной.

22. Известно, что в 1 г живой клетчатки (например, дерева) содержится $7.5 \cdot 10^{10}$ ядер радиоактивного углерода. После гибели организма (дерева) их концентрация уменьшается по закону $N = N_0 e^{-\lambda t}$,

где N_0 – начальная концентрация; $\lambda = \frac{\ln 2}{T}$; T – период полураспада (для радиоактивного углерода равен 5570 годам).

Построить таблицу зависимости концентрации радиоактивных ядер от времени для интервала времени от 0 до 6000 лет с шагом 500 лет, считая за 0 момент гибели организма (дерева).

23. Плотность воздуха убывает с высотой по закону $\rho = \rho_0 e^{-hz}$. Считая, что $\rho_0 = 1.29 \text{ кг/м}^3$, $z = 1.25 \cdot 10^{-4} \text{ 1/м}$, напечатать таблицу зависимости плотности от высоты значений от 0 до 1000 м с шагом 100 м. Определить, на какой высоте плотность будет меньше 1 кг/м³.

24. Концентрация хлорной извести в бассейне объемом $V \text{ м}^3$ составляет 10 г/л . Через одну трубу в бассейн вливают чистую воду с объемной скоростью $Q \text{ м}^3/\text{час}$, через другую трубу с такой же скоростью вода выливается. При условии идеального перемешивания концентрация хлорной извести изменяется по закону

$$C = C_0 e^{-Qt/V},$$

где t – время; C_0 – начальная концентрация.

Определить, через какое время концентрация хлорной извести достигнет безопасной для человека величины $0,1 \text{ г/л}$. Задачу решить при $Q=150 \text{ м}^3/\text{час}$, $V=10000 \text{ л}$, $C_0=10 \text{ г/л}$.

25. В 2000 году урожай ячменя составил 20 ц с га . В среднем каждые два года за счет применения передовых агротехнических приемов урожай увеличивается на 5% . Определить, через сколько лет урожайность достигнет 25 ц с га .

26. Два спортсмена одновременно начинают движение из одной точки. Первый спортсмен начинает движение со скоростью 10 км/час и в начале каждого следующего часа увеличивает скорость на 1 км/час . Вторым начинает движение со скоростью 9 км/час и в начале каждого следующего часа увеличивает скорость на $1,6 \text{ км/час}$. Выяснить какой спортсмен преодолеет больший путь через 1 час ; через 4 часа . Определить, когда второй спортсмен догонит первого.

27. Подрабатывая вечерами курьером, студент решил накопить сумму в S рублей для покупки компьютера. В первый месяц он отложил P рублей. Затем его вклад возрастал на k процентов больше предыдущего вклада. Через какое число месяцев студент сможет купить компьютер?

28. В водоеме 100 тонн рыб. Каждый год рыболовецкая бригада вылавливает 15 тонн . Воспроизводство рыбы – 5% в год. Для сохранения воспроизводства рыбы необходимо прекращать лов, когда в водоеме ее останется не менее 5 тонн . Через какое число лет должен быть прекращен лов рыбы?

29. Математик Кеплер обратил внимание на то, что с увеличением номеров элементов числового ряда Фибоначчи отношение предыдущего элемента к последующему приближается к отношению золотого сечения ($0,618$). Записать программу проверки этого утверждения.

30. Записать программу, помогающую вкладчику узнать: сколько лет нужно ожидать сумму S , если начальный вклад равен V при P процентах прироста вклада за год?

31. Задача Л.Эйлера. Некий чиновник купил лошадей и быков на 1770 талеров . За каждую лошадь он уплатил по 31 талеру , а за каждого быка – по 21 талеру . Сколько лошадей и быков купил чиновник? Выведите все варианты.

Лабораторная работа № 5

РАБОТА С ОДНОМЕРНЫМИ МАССИВАМИ

Создание и заполнение массивов

Массивы – это группа элементов одинакового типа (*double*, *float*, *int* и т.п.). Из объявления массива компилятор должен получить информацию о типе элементов массива и их количестве. Объявление массива имеет два формата:

спецификатор-типа *описатель* [*константное_выражение*];
спецификатор-типа *описатель* [];

Описатель – это идентификатор массива.

Спецификатор-типа задает тип элементов объявляемого массива. Элементами массива не могут быть функции и элементы типа *void*.

Константное-выражение в квадратных скобках задает количество элементов массива. *Константное-выражение* при объявлении массива может быть опущено в следующих случаях:

- при объявлении массив инициализируется;
- массив объявлен как формальный параметр функции;
- массив объявлен как ссылка на массив, явно определенный в другом файле проекта.

В языке C/C++ определены только одномерные или линейные массивы, т.е. за каждым элементом массива закреплен только один его порядковый номер.

Описывать массив *int A[10]* – это значит выделить 10 последовательных элементов в памяти (каждый размером *sizeof(int)*) для массива с именем *A*, элементы которого имеют тип *int*. Каждый элемент массива в общем виде описывается как *A[i]*, где *A* – имя массива; *i* – номер или индекс массива ($0 \leq i < n$); *A[i]* – значение элемента массива.

Отметим, что в языке C/C++ первый элемент массива имеет индекс равный 0.

Пример описания массива:

```
double b[10]; /* вектор из 10 элементов имеющих тип double */
```

После объявления массива каждый его элемент можно обработать, указав идентификатор (имя) массива и индекс элемента в квадратных скобках. Например, запись *Mas[2]*, *VectorZ[10]* позволяет обратиться ко второму элементу массива *Mas* и десятому элементу массива *VectorZ*.

Задача 5.1. Заполнить массив 20 целыми числами. Элементы массива вывести на экран монитора.

Листинг программы

```
#include <stdio.h>
#include <mem.h>
int main()
{
    //Заполнение одномерного массива
```

```

int a[20];
//Input
for(int i=0;i<20;i++)
{
    printf("Input a[%d] ", i);
    scanf("%d", &a[i]);
}
//Вывод элементов массива в строку
for(int i=0;i<20;i++)
    printf("%d ", a[i]);
printf("\n");
//Вывод элементов массива в столбец
for(int i=0;i<20;i++)
    printf("%d\n", a[i]);
    getchar();getchar();
return 0;
}

```

Вариант работы программы с массивом из 3 элементов целого типа представлен на рис. 5.1.

```

C:\Program Files\Borland\CBuilder6\Projects\Project2.exe
Input a[0] 4
Input a[1] -2
Input a[2] 3

```

Рис. 5.1

Заполнение одномерного массива с помощью генератора случайных чисел

Функция `random(100)` генерирует случайные числа от 0 до 99 целого типа.

Примеры:

//Генерируются числа от 0 до 99 целого типа

a[i]=random(100);

//Генерируются целые числа в диапазоне от -50 до 49

a[i]=random(100)-50;

Для генерации целых чисел в диапазоне $[A, B]$ можно использовать следующее выражение: $random(B-A+1)+A$, а для генерации вещественных – $(double)rand()/RAND_MAX*(B-A)+A$.

Для того чтобы при каждом запуске программа генерировала новые значения, необходимо добавить вызов функции `randomize()`.

Задача 5.2. Заполнить массив N целыми числами от -15 до 38 . Элементы массива вывести в строку.

Листинг программы

```
#include <stdio.h>
int main()
{ int a[100];
  randomize();
  int k;
  printf("Vvedite razmernost massiva=");
  scanf("%d",&k);
  //Заполнение массива
  for(int i=0;i<k;i++)
    a[i]=random(38-(-15)+1)-15;
  //Вывод элементов массива в строку
  for(int i=0;i<k;i++)
    printf("a[%d]=%d ", i,a[i]);
  printf("\n");
  getchar();getchar();
  return 0;
}
```

Нахождение суммы (или произведения) элементов

Задача 5.3. Заполнить массив 15 целыми случайными числами от – 10 до 15. Найти сумму четных чисел.

Листинг программы

```
#include <stdio.h>
int main()
{ int a[15];
  randomize();
  int s;
  s=0;
  for(int i=0;i<15;i++)
  {
    a[i]=random(15-(-10)+1)-10;
    printf("a[%d]=%d ", i,a[i]);
    if (a[i]%2==0) s=s+a[i];
  }
  printf("\n");
  printf("s=%d", s);
  getchar();getchar();
  return 0;
}
```

Результаты работы программы приведены на рис. 5.2.

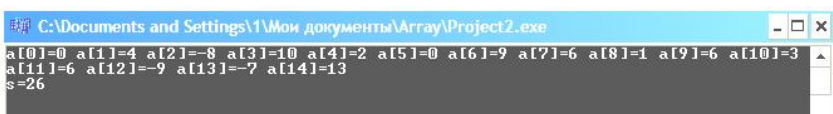


Рис. 5.2

Задача 5.4. Дан массив целых чисел. Найти минимальный элемент и его номер. Вывести элементы массива в порядке убывания.

Листинг программы

```
#include <stdio.h>
int main()
{
    const n=15;
    int a[n]={3,5,2,8,1,-5,4,-2,-13,-8,2,3,1,-3,4};
    printf("First massiv\n");
    //Вывод исходного массива
    for (int i=0; i<n; i++)
        printf("a[%d]=%4d ", i, a[i]);
    printf("\n");
    //Поиск минимального числа и его индекса
    int MinValue=a[0]; int MinIndex=0;
    for (int i=1; i<n; i++)
        if (a[i]<MinValue)
        {
            MinValue=a[i];
            MinIndex=i;
        }
    printf("Min=%d\n", MinValue);
    printf("MinIndex=%d\n", MinIndex);
    //Сортировка
    int tmp;
    for (int i=0; i<n-1; i++)
        for (int j=i+1; j<n; j++)
            if (a[i]>a[j])
            {
                tmp=a[i]; a[i]=a[j]; a[j]=tmp;
            }
    printf("Massiv posle sortirovki\n");
    for (int i=0; i<n; i++)
        printf("%4d", a[i]);
    getchar();
    return 0;
}
```

Результаты работы программы приведены на рис. 5.3.

```
First massiv
a[0]= 3 a[1]= 5 a[2]= 2 a[3]= 8 a[4]= 1 a[5]= -5 a[6]= 4 a[7]= -2
a[8]= -13 a[9]= -8 a[10]= 2 a[11]= 3 a[12]= 1 a[13]= -3 a[14]= 4
Min=-13
Min=8
Massiv after sortirovka
-13 -8 -5 -3 -2 1 1 2 2 3 3 4 4 5 8_
```

Рис. 5.3

Задание для индивидуального выполнения
по теме «Работа с одномерными массивами»

Задание И5.1. Элементы массива генерируются случайным образом в заданном диапазоне $[a; b]$ согласно варианту.

- | | | |
|-----|------------|------------|
| 1. | $a = -10$ | $b = 7;$ |
| 2. | $a = -15$ | $b = 37;$ |
| 3. | $a = 25$ | $b = 57;$ |
| 4. | $a = -65$ | $b = 2;$ |
| 5. | $a = -100$ | $b = 107;$ |
| 6. | $a = -15$ | $b = 88;$ |
| 7. | $a = -18$ | $b = 23;$ |
| 8. | $a = -37$ | $b = 56;$ |
| 9. | $a = -22$ | $b = 132;$ |
| 10. | $a = -8$ | $b = 45;$ |
| 11. | $a = -19$ | $b = 37;$ |
| 12. | $a = -48$ | $b = 32;$ |
| 13. | $a = -13$ | $b = 10;$ |
| 14. | $a = -105$ | $b = 66;$ |
| 15. | $a = -25$ | $b = 87;$ |
| 16. | $a = -78$ | $b = 43;$ |
| 17. | $a = -16$ | $b = 21;$ |
| 18. | $a = -45$ | $b = 52;$ |
| 19. | $a = -1$ | $b = 48;$ |
| 20. | $a = -9$ | $b = 60;$ |
| 21. | $a = -3$ | $b = 29;$ |
| 22. | $a = -74$ | $b = 39;$ |
| 23. | $a = -23$ | $b = 20;$ |
| 24. | $a = -12$ | $b = 4;$ |
| 25. | $a = -66$ | $b = 33;$ |
| 26. | $a = -10$ | $b = 29;$ |
| 27. | $a = 20$ | $b = 44;$ |
| 28. | $a = -11$ | $b = 31;$ |
| 29. | $a = -5$ | $b = 5;$ |
| 30. | $a = -23$ | $b = 35;$ |
| 31. | $a = -17$ | $b = 16.$ |

При решении всех задач использовать структурированный тип – массив.

1. Дана последовательность целых чисел $a_0, \dots, a_{50}$. Получить сумму тех чисел данной последовательности, которые кратны 5, нечетны и отрицательны, удовлетворяют условию $a_i < i^2$.

2. Даны натуральное число n и последовательность целых чисел $a_0, \dots, a_{n-1}$. Найти количество и сумму тех элементов данной последовательности, которые делятся на 5 и не делятся на 7.

3. Даны натуральные числа n, p и последовательность целых чисел $a_0, \dots, a_{n-1}$. Получить произведение элементов данной последовательности, кратных p .

4. Даны натуральные p, q и последовательность целых чисел $a_0, \dots, a_{66}$ ($p > q \geq 0$). В последовательности $a_0, \dots, a_{66}$ заменить на нуль все элементы, модуль которых при делении на p дает в остатке q .

5. Даны натуральное число n , последовательность действительных чисел $a_0, \dots, a_{n-1}$. Получить удвоенную сумму всех положительных элементов последовательности $a_0, \dots, a_{n-1}$.

6. Даны натуральное число n , последовательность натуральных чисел $a_0, \dots, a_{n-1}$. Вычислить обратную величину произведения тех элементов a_i последовательности $a_0, \dots, a_{n-1}$, для которых выполнено $i+1 < a_i < i!$.

7. Даны натуральное число n , последовательность действительных чисел $a_0, \dots, a_{n-1}$. В последовательности $a_0, \dots, a_{n-1}$ все отрицательные элементы увеличить на 0,5, а все неотрицательные заменить на 0,1.

8. Даны натуральное число n , последовательность целых чисел $a_0, \dots, a_{n-1}$. В последовательности $a_0, \dots, a_{n-1}$ все элементы меньше двух заменить нулями. Получить сумму элементов, принадлежащих отрезку $[3; 7]$, а также число таких элементов.

9. Даны натуральное число n , последовательность целых чисел $a_0, \dots, a_{n-1}$. В последовательности $a_0, \dots, a_{n-1}$ все неотрицательные элементы, не принадлежащие отрезку $[1; 2]$, заменить на единицу. Получить число отрицательных элементов и число элементов, принадлежащих отрезку $[1; 2]$.

10. Даны натуральное число n , целые числа $a_0, \dots, a_{n-1}$. Получить сумму положительных и число отрицательных четных элементов последовательности $a_0, \dots, a_{n-1}$.

11. Даны натуральное число n , последовательность целых чисел $a_0, \dots, a_{n-1}$. Заменить все большие семи элементы последовательности $a_0, \dots, a_{n-1}$ числом семь. Вычислить количество таких элементов, которые имеют четный индекс.

12. Дана последовательность целых чисел $a_0, \dots, a_{45}$. Получить число отрицательных элементов последовательности $a_0, \dots, a_{35}$ и число нулевых элементов всей последовательности.

13. Даны натуральное число n , целое a и последовательность целых чисел $x_0, \dots, x_{n-1}$. Если в последовательности $x_0, \dots, x_{n-1}$ есть хотя бы один элемент, равный a , то получить сумму всех элементов, следующих за первым таким элементом, в противном случае ответом должно быть число -10 .

14. Даны целые числа $a_0, \dots, a_{49}$. Получить последовательность $b_0, \dots, b_{49}$, которая отличается от исходной последовательности тем, что все нечетные элементы удвоены.

15. Даны натуральное число n , действительные числа r и последовательность действительных чисел $a_0, \dots, a_{n-1}$ ($n \geq 2$). Сколько среди точек $(a_0, a_{n-1}), (a_1, a_{n-2}), \dots, (a_{n-1}, a_0)$ таких, которые принадлежат кругу радиуса r с центром в начале координат.

16. Дан одномерный массив. Переставить в обратном порядке элементы массива, расположенные между первым минимальным и первым максимальным элементами.

17. Дан массив целых чисел, состоящих из 20 элементов. Найти сумму элементов, имеющих нечетное значение. Вывести индексы тех элементов, значения которых больше заданного числа A . Определить, есть ли в заданном массиве положительные элементы, кратные K (K вводится с клавиатуры). Если таковых нет, то вывести сообщение.

18. Дан массив целых чисел. Найти, сколько в нем пар одинаковых соседних элементов.

19. Дан массив целых чисел, состоящий из 25 элементов. Найти:

- сумму элементов, имеющих нечетные индексы;
- подсчитать количество элементов массива, значения которых больше заданного числа A и кратных пяти;
- найти номер первого отрицательного элемента, делящегося на пять с остатком.

20. Дан массив чисел (не менее 100 элементов). Найти сумму пяти наименьших элементов.

21. Заданы N целых чисел. Определить, сколько среди них четных и притом делящихся на семь без остатка. Вывести эти числа.

22. Написать программу, которая перемещает последний элемент массива, вставляя его, после первого отрицательного элемента этого же массива.

23. Все элементы массива, значения которых кратны трем, заменить числом 13.

24. Дан массив целых чисел. Найти сумму семи наибольших элементов.

25. Заданы действительное число X и массив $A[n]$. Определить элемент, который по значению ближайший к числу X . Напечатать элемент и его номер.

26. В заданном массиве вывести все максимальные элементы и их номера.

27. В массиве вещественных чисел изменить знак отрицательных чисел, после этого определить максимальный по величине элемент массива и его номер в массиве.

28. Задан массив целых чисел. Определить количество отрицательных чисел и среднее арифметическое положительных чисел.

29. Задан массив целых чисел. Определить среднее геометрическое чисел, кратных пяти.

30. Задан массив целых чисел. Найти два минимальных элемента и поменять их местами.

31. Задан массив целых чисел. Определить среднее арифметическое элементов массива, которые находятся между максимальным и минимальным элементами включительно.

Задание И5.2.

1. Дан массив из 25 вещественных чисел. Записать в этот же массив сначала все положительные числа и нули, а затем все отрицательные, сохраняя порядок их следования.

2. Дана последовательность действительных чисел $A_0, \dots, A_{49}$. Получить последовательность $B = \{A_1, A_2, \dots, A_{49}, A_0\}$.

3. Дан массив действительных чисел $A[n]$. Получить массив $B[n]$, где $B = \{A_0, A_0 + A_1, A_0 + A_1 + A_2, \dots, A_0 + A_1 + \dots + A_{n-1}\}$.

4. Дан массив $A[n]$. Получить новый массив, в котором:

$$B[i] = \begin{cases} e^{A[i]} + e^{-A[i]}, & \text{если } A[i] > 0; \\ 11, & \text{если } A[i] = 0; \\ e^{A[i]} - e^{-A[i]}, & \text{если } A[i] < 0. \end{cases}$$

5. Дана последовательность целых чисел $A_0, \dots, A_{49}$. Получить последовательность $B_0, \dots, B_{49}$, которая отличается от исходной тем, что все нечетные элементы удвоены.

6. Даны натуральное число n , действительные числа $X_0, \dots, X_{n-1}$. В последовательности $X_0, \dots, X_{n-1}$ все элементы, которые меньше двух, заменить нулями. Подсчитать их количество.

7. Даны натуральное число n , действительные числа $A_0, \dots, A_{n-1}$. В последовательности $A_0, \dots, A_{n-1}$ все отрицательные элементы увеличить в два раза, а все неотрицательные заменить на 10.

9. Даны целые числа $A, N, X_0, \dots, X_{n-1}$. Определить, каким по счету идёт в последовательности $X_0, \dots, X_{n-1}$ элемент, равный A . Если такого элемента нет, то ответом должно быть число 0.

10. Дан массив действительных чисел $A[30]$. Получить новый массив $D[15] = \{A_1 \cdot A_6, A_2 \cdot A_{17}, A_3 \cdot A_{18}, \dots, A_{15} \cdot A_{30}\}$.

11. Задан целочисленный массив $B[20]$. Определить максимальный и минимальный элементы массива и их порядковые номера.

12. Задан массив $B[15]$. Найти максимальный и минимальный элементы массива и поменять их местами.

13. Определить сумму элементов массива $A[25]$, значения которых кратны трем и подсчитать их количество.

14. Для заданной последовательности B из n элементов найти сумму $C = \sum_{k=0}^{n-1} \frac{(B_k + 1)}{K}$.

15. Для заданной последовательности X из n элементов определить отношение A/B , где $A = \prod_{k=0}^{n-1} X_k$, $B = \sum_{k=0}^{n-1} X_k$.

16. Найти сумму 1-го, 4-го, 9-го, 16-го, ..., k -го элементов одномерного массива X из n элементов (k – наибольшее целое число, не превышающее $\sqrt{n}$).

17. При заданных элементах $X_0, X_1, \dots, X_{n-1}$ и четном n найти значения сумм $C = \sum_{i=0}^{\frac{n}{2}-1} X_i$, $D = \sum_{i=\frac{n}{2}}^n X_i$.

18. Найти произведение, сомножители которого представлены каждым третьим элементом: X_0, X_3, X_6 , и т. д. – заданного массива $X_0, X_1, \dots, X_{n-1}$.

19. При заданных элементах $X_0, X_1, \dots, X_{n-1}$ и четном n найти разность сумм $C1 = \sum_{i=0}^{\frac{n}{2}-1} X_{2 \cdot i}$, $C2 = \sum_{i=0}^{\frac{n}{2}-1} X_{2 \cdot i + 1}$.

20. При заданных координатах $C_0, C_1, \dots, C_{n-1}$ одной и координатах $B_0, B_1, \dots, B_{n-1}$ другой точки n -мерного пространства найти расстояние между ними по формуле: $R = \sqrt{(C_0 - B_0)^2 + \dots + (C_{n-1} - B_{n-1})^2}$.

21. Дано натуральное число n . Получить последовательность $B_0, \dots, B_{n-1}$, где значение $B_i = i!$.

22. При заданных коэффициентах $A_0, A_1, \dots, A_{n-1}$ и заданном значении X вычислить значение многочлена $A_0X + A_1X^2 + \dots + A_{n-1}X^n$.

24. При заданных элементах массива X : $X_0, X_1, \dots, X_{n-1}$ найти по формуле $Y_k = X_0X_k + 1$ последовательность элементов массива Y : $Y_0, Y_1, \dots, Y_{n-1}$.

25. Из одномерного массива X из n элементов сформировать массив T по правилу: $T_0 = X_0$, $T_1 = \sqrt{|X_1|}$, ..., $T_{n-1} = \sqrt[n]{|X_{n-1}|}$.

26. Дана последовательность чисел, среди которых имеется один ноль. Вывести на печать все числа, включительно до нуля.

28. Дан целочисленный массив с количеством элементов N . Напечатать те элементы, индексы которых являются степенями двойки.

30. Даны координаты N точек на плоскости. Найти номера пары точек, расстояние между которыми наибольшее.

31. Дан одномерный массив $A[n]$. Найти:

$$S = \max(A_0, A_2, \dots, A_{2k}, \dots) + \min(A_1, A_3, \dots, A_{2k+1}, \dots).$$

Лабораторная работа № 6 МНОГОМЕРНЫЕ МАССИВЫ

Поскольку элементом массива может быть массив, можно определить и многомерные массивы. Они формализуются списком константных-выражений, следующих за идентификатором массива, причем каждое константное-выражение заключается в свои квадратные скобки.

Каждое константное-выражение в квадратных скобках определяет число элементов по данному измерению массива, так что объявление двумерного массива содержит два константных-выражения, трехмерного – три и т.д.

Примеры:
int a[2][3]; / представлено в виде матрицы*
a[0][0] a[0][1] a[0][2]
*a[1][0] a[1][1] a[1][2] */*
int w[3][3] = { { 2, 3, 4 },
{ 3, 4, 8 },
{ 1, 0, 9 } };

В последнем примере объявлен массив *w[3][3]*. Списки, выделенные в фигурные скобки, соответствуют строкам массива, в случае отсутствия скобок инициализация будет выполнена неправильно.

В языке C/C++ можно использовать сечения массива, однако на использование сечений накладывается ряд ограничений. Сечения формируются вследствие опускания одной или нескольких пар квадратных скобок. Пары квадратных скобок можно отбрасывать только справа налево и строго последовательно. Сечения массивов используются при организации вычислительного процесса в функциях языка C/C++ , разрабатываемых пользователем.

Примеры:
int s[2][3];

Если при обращении к некоторой функции написать *s[0]*, то будет передаваться нулевая строка массива *s*.

int b[2][3][4];

При обращении к массиву *b* можно написать, например, *b[1][2]* и будет передаваться вектор из четырех элементов, а обращение *b[1]* даст двумерный массив размером 3 на 4. Нельзя написать *b[2][4]*, подразумевая, что передаваться будет вектор, потому что это не соответствует ограничению, наложенному на использование сечений массива.

Пример объявления символьного массива.
char str[] = "объявление символьного массива";

Следует учитывать, что в символьном литерале находится на один элемент больше, так как последний из элементов является управляющей последовательностью *'\0'*.

Индексное выражение задает элемент массива и имеет вид:
Выражение1 [Выражение2]

Тип индексного выражения является типом элементов массива, а значение представляет величину, адрес которой вычисляется с помощью значений *Выражение1* и *Выражение2*.

Обычно *Выражение1* – это указатель, например, идентификатор массива, а *Выражение2* – это целая величина. Однако требуется только, чтобы одно из выражений было указателем, а второе целочисленной величиной. Поэтому *Выражение1* может быть целочисленной величиной, а *Выражение2* указателем. В любом случае *Выражение2* должно быть заключено в квадратные скобки. Хотя индексное выражение обычно используется для ссылок на элементы массива, тем не менее, индекс может появляться с любым указателем.

Индексные выражения для ссылки на элементы одномерного массива вычисляются путем сложения целой величины со значениями указателя с последующим применением к результату операции разадресации (*).

Так как одно из выражений, указанных в индексном выражении, является указателем, то при сложении используются правила адресной арифметики, согласно которым целая величина преобразуется к адресному представлению путем умножения ее на размер типа, адресуемого указателем. Пусть, например, идентификатор *arr* объявлен как массив элементов типа *double*.

double arr[10];

Таким образом, чтобы получить доступ к *i*-му элементу массива *arr*, можно написать *arr[i]*, что, в силу сказанного выше, эквивалентно *i[arr]*. При этом величина *i* умножается на размер типа *double*, а затем это значение складывается со значением указателя *arr*, что, в свою очередь, дает адрес *i*-го элемента массива. К полученному адресу применяется операция разадресации, т.е. осуществляется выборка элемента массива *arr* по сформированному адресу.

Таким образом, результатом индексного выражения *arr[i]* (или *i[arr]*) является значение *i*-го элемента массива.

Для ссылки на элемент многомерного массива индексное выражение должно иметь несколько индексов, заключенных в квадратные скобки:

Выражение1 [Выражение2][Выражение3]...

Такое индексное выражение интерпретируется слева направо, т.е. вначале рассматривается первое индексное выражение:

Выражение1 [Выражение2]

Результат этого выражения есть адресное выражение, с которым складывается *Выражение3* и т.д. Операция разадресации осуществляется после вычисления последнего индексного выражения. Отметим, что операция разадресации не применяется, если значение последнего указателя адресует величину типа массива.

Пример:

```
int mass [2][5][3];
```

Рассмотрим процесс вычисления индексного выражения `mass[1][2][2]`.

1. Вычисляется выражение `mass[1]`. Ссылка индекс 1 умножается на размер элемента этого массива, элементом же этого массива является указатель на двумерный массив, содержащий 5×3 элементов, имеющих тип `int`. Получаемое значение складывается со значением указателя `mass`. Результатом является указатель на второй двухмерный массив размером (5×3) в трехмерном массиве `mass`.

2. Второй индекс 2 указывает на массив из трех элементов типа `int` и складывается с адресом, соответствующим `mass[1]`.

3. Так как каждый элемент трехмерного массива – это величина типа `int`, то индекс 2 увеличивается на размер типа `int` перед сложением с адресом `mass[1][2]`.

4. Наконец, выполняется разадресация полученного указателя. Результирующим выражением будет элемент типа `int`.

Если было бы указано `mass[1][2]`, то результатом был бы указатель на массив из трех элементов типа `int`. Соответственно значением индексного выражения `mass[1]` является указатель на двухмерный массив.

Задача 6.1. Ввод-вывод, обнуление. Р

Листинг программы

```
#include <stdio.h>
#include <mem.h>
int main()
{
    //Two-dimensional array
    int b[2][2];
    //Input
    for(int i=0;i<2;i++)
        for(int j=0;j<2;j++)
        {
            printf("Input b[%d][%d] ", i, j);
            scanf("%d", &b[i][j]);
        }
    //Output as matrix
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
            printf("%4d", b[i][j]);
        printf("\n");
    }
    //Zeroing
    memset(b, 0, sizeof(b));
    getchar();getchar();
    return 0;
}
```

Результат работы программы задачи 6.1 показан на рис. 6.1.

```
Input  b[0][0] -4
Input  b[0][1] 2
Input  b[1][0] 0
Input  b[1][1] 1
-4    2
0     1
```

Рис. 6.1

Задача 6.2. Олимпиада по информатике проводится в m туров. Вывести победителей каждого тура и победителей в общем зачете. Максимальное число баллов в туре – 50.

Листинг программы

```
#include <stdio.h>
int main()
{
    const n=3; //число участников
    const m=2; //число туров
    unsigned char rez[n][m+1]; //последний столбец на сумму
    unsigned char max[m+1]; //лучшие результаты
    char name[n][20]; //фамилии
    int i, j;
    //Ввод данных
    for (i=0; i<n; i++)
    {
        printf("Input a surname of %d-st participant ", i+1);
        scanf("%s", name[i]);
        for (j=0; j<m; j++)
        {
            printf("Input its result in %d-st round ", j+1);
            scanf("%d", &rez[i][j]);
        }
    }
    //Подсчет суммарного результата
    for (i=0; i<n; i++)
    {
        rez[i][m]=0;
        for (j=0; j<m; j++)
            rez[i][m]+=rez[i][j];
    }
    //Поиск максимальных результатов
    for (j=0; j<m+1; j++)
    {
        max[j]=rez[0][j];
        for (i=1; i<n; i++)
            if (rez[i][j]>max[j]) max[j]=rez[i][j];
    }
}
```

```

//Вывод победителей в турах
int t;
for (j=0; j<m; j++)
{
    printf("Winners in %d-st round with result %d\n", j+1, max[j]);
    t=0;
    for (i=0; i<n; i++)
        if (rez[i][j]==max[j])
        {
            t++;
            printf("%d. %-s\n", t, name[i]);
        }
}
//Вывод победителя в общем зачете
printf("Winners in the general offset with result %d\n", max[m]);
t=0;
for (i=0; i<n; i++)
    if (rez[i][m]==max[m])
    {
        t++;
        printf("%d. %-s\n", t, name[i]);
    }
getchar();getchar();
return 0;
}

```

Результат работы программы задачи 6. показан на рис. 6.2.

```

Input a surname of 1-st participant Ivanov
Input its result in 1-st round 23
Input its result in 2-st round 34
Input a surname of 2-st participant Petrov
Input its result in 1-st round 45
Input its result in 2-st round 41
Input a surname of 3-st participant Sidorov
Input its result in 1-st round 35
Input its result in 2-st round 47
Winners in 1-st round with result 45
1. Petrov
Winners in 2-st round with result 47
1. Sidorov
Winners in the general offset with result 86
1. Petrov
_

```

Рис. 6.2

Задание для индивидуального выполнения по теме «Работа с многомерными массивами»

Задание ИБ.1. Работа с двумерными массивами (матрицами).

1. Задана квадратная матрица размером $k \times k$, составить программу вычисления суммы и произведения элементов матрицы, находящихся на главной диагонали.

2. Составить программу, которая позволяет для матрицы размерностью $m \times n$ вычислить произведение положительных, сумму отрицательных и количество нулевых элементов.

3. Двумерный массив C содержит пять строк и столбцов, одномерный массив A содержит пять элементов. Вычислить произведение матрицы на вектор.

4. Дана матрица B . Для каждого столбца с чётным номером вычислить и напечатать сумму квадратов элементов этого столбца, а для каждого столбца с нечётным номером вычислить произведение элементов.

5. Найти минимальный элемент j -го столбца матрицы A размером $n \times n$, для которого сумма абсолютных значений элементов максимальна (если таких столбцов несколько, взять первый из них).

6. Дан двумерный массив A . Каждый элемент массива, стоящий выше главной диагонали, заменить его квадратом, а ниже диагонали – кубом его значения. Элементы главной диагонали оставить без изменения.

7. Задан двумерный массив B . Найти в нём максимальный элемент и разделить на него каждый элемент массива B .

8. Заданы матрица $n \times n$ и число $k < n$. Строку с максимальным по модулю элементом в k -м столбце переставить с k -й строкой.

9. Дана матрица B . Для каждого столбца матрицы вычислить и напечатать разность между квадратом суммы и суммой квадратов элементов этого столбца.

10. Заданы двумерный массив 5×5 и число K . Разделить элементы K -й строки на диагональный элемент, расположенный в данной строке.

11. Определить и напечатать среднее значение элементов двумерного массива размером $n \times m$. Найти индексы элемента, наиболее близкого по значению к среднему.

12. Составить программу нахождения минимального положительного элемента матрицы размерностью $m \times n$, а также значение индексов этого элемента.

13. Составить программу для определения количества отрицательных, положительных и равных нулю элементов матрицы $R(m \times n)$.

14. Составить программу вычисления суммы и произведения элементов квадратной матрицы A размерностью $n \times n$, расположенных ниже главной диагонали.

15. Составить программу для преобразования квадратной матрицы, состоящего в симметричной относительно главной диагонали перестановки её элементов.

16. Дана действительная матрица размера $m \times n$. Найти значение наибольшего по модулю элемента матрицы, а также индексы всех элементов с найденными значениями модуля.

17. Дана действительная матрица размера $m \times n$. Найти максимальные элементы по строкам и их сумму.

18. В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.

19. В данной действительной матрице размера 6×9 поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением. Предполагается, что эти элементы единственны.

20. Дана действительная матрица размера $m \times n$, все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.

21. Дана целочисленная квадратная матрица порядка восемь. Найти наименьшее значение из элементов столбца, который обладает наибольшей суммой модулей элементов. Если таких столбцов несколько, то взять первый.

22. Даны натуральное число n , целочисленная квадратная матрица порядка n . Получить $b_0, b_1, \dots, b_{n-1}$, где b_i – это наименьшее значение элементов, находящихся в начале i -й строки матрицы до элемента, принадлежащего главной диагонали включительно.

23. Дана целочисленная квадратная матрица порядка n . Получить $b_0, b_1, \dots, b_{n-1}$, где b_i – это значение первого по порядку положительного элемента i -й строки (если таких элементов нет, то принять $b_i = 1$).

24. Дана целочисленная квадратная матрица порядка n . Получить $b_0, b_1, \dots, b_{n-1}$, где b_i – это сумма элементов, расположенных за первым отрицательным элементом в i -й строке (если все элементы неотрицательны, то принять $b_i = 100$).

25. Дана целочисленная квадратная матрица порядка n . Получить $b_0, b_1, \dots, b_{n-1}$, где b_i – это сумма элементов, предшествующих последнему отрицательному элементу i -й строки (если все элементы строки неотрицательны, то принять $b_i = -1$).

26. Дана действительная квадратная матрица порядка n . Отсортировать в порядке убывания те строки, которые начинаются с отрицательного элемента.

27. Дана матрица $A[m, m]$ и массив $X[m]$. Нечетные строки матрицы заменить на $X[m]$, четные столбцы матрицы A заменить на $X[m]$.

28. Дана матрица $A[m, m]$. В матрице A поменять местами 1-ю и 2-ю строки, 3-ю и 4-ю и т.д.

29. Дана матрица $A[m, m]$ и натуральное число K . Получить из массива A массив B , заменив все элементы K -го столбца на нулевые.

30. Заданный двумерный массив преобразовать в одномерный.

31. Заданный одномерный массив преобразовать в двумерный.

Задание И6.2. Вложенные циклы в матричных задачах.

В решении задачи можно использовать произвольный способ ввода и вывода как матрицы, так и результатов.

1. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Отобрать кандидатов на олимпиады (с отличными оценками) по каждому из предметов.

2. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Найти среднюю в группе оценку по каждому из предметов.

3. Спортсмены на соревнованиях совершают 6 попыток в прыжках в длину. Определить лучший результат для каждого участника.

4. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Найти в группе среднюю оценку для каждого учащегося.

5. Для группы фирм известен курс их акций за каждый из месяцев календарного года. Составить список тех фирм, курс акций которых все время повышался (т.е. курс за каждый последующий месяц больше, чем за предыдущий).

6. Выступления N спортсменов оцениваются M судьями по одной и той же числовой шкале. Нужно узнать конечный результат каждого спортсмена, если он вычисляется так: из всех M оценок удаляются максимальная и минимальная (если таких оценок несколько, то удаляется только одна), затем из оставшихся $(M-2)$ находится их среднее арифметическое.

7. Известна среднемесячная температура воздуха на следующих островах Карибского моря: Куба, Тринидад, Ямайка, Гаити. Определить, на каком из островов среднегодовая температура максимальна.

8. Известна среднемесячная температура воздуха на следующих островах Карибского моря: Куба, Тринидад, Ямайка, Гаити. Определить для каждого из месяцев, на каком из островов максимальная температура.

9. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждого четного столбца максимальный элемент. Найти произведение этих элементов.

10. Дан двумерный массив из 5 строк и 6 столбцов. Определить, какая строка массива имеет максимальную сумму элементов (считать, что строка с одинаковой суммой нет).

11. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждой нечетной строки минимальный элемент. Найти произведение этих элементов.

12. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждой строки минимальный элемент. Среди этих элементов найти максимальный.

13. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждого столбца максимальный элемент. Среди этих элементов найти минимальный.

14. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждого столбца массива сумму минимального и максимального элементов. Найти произведение этих сумм.

15. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждой строки массива произведение минимального и максимального элементов. Найти сумму этих произведений.

16. Дан двумерный массив из 5 строк и 6 столбцов. Определить для каждой строки массива произведение элементов. Найти, в какой строке это произведение максимально.

17. Дан двумерный массив из 5 строк и 6 столбцов. Определить, какой столбец массива имеет минимальную сумму элементов (считать, что столбцов с одинаковой суммой нет).

18. Известны места 8 лыжников на каждом из 7 стартов Кубка мира. Определить победителя в общем зачете (с минимальной суммой мест). Если таких несколько, то победителем считается тот, кто лучше других претендентов на первое место выступил на последнем этапе.

19. По результатам метеорологических наблюдений за 10 последних лет известно количество солнечных дней в году для пяти морских курортов. Кроме этого известны расстояния до них. Определить курорт с наиболее благоприятным климатом (с максимальным суммарным количеством солнечных дней за время наблюдений). Если таких несколько, то вывести ближайший из них.

20. В автопарке находится 10 автомобилей. Известен их пробег в течение каждого из 5 рабочих дней. Определить, какой из автомобилей за рабочую неделю преодолел максимальное расстояние. Если таких несколько, то вывести хотя бы одного из них.

21. Бригада рабочих состоит из 8 человек. Известно сколько деталей выпустил каждый из них в течение каждого из 5 рабочих дней. Определить, какой рабочий произвел максимальное количество деталей. Если таких несколько, то вывести хотя бы одного из них.

22. На каждом этаже трехэтажного дома жилых 6 комнат, каждая из которых имеет форму прямоугольника. Длина и ширина каждой комнаты известны. Определить, какой из этажей дома имеет минимальную жилую площадь. Если таких несколько, то вывести хотя бы одного из них.

23. Известны итоги сдачи экзаменов группой из 20 студентов по четырем предметам. Определить, по результатам каких экзаменов в группе максимальный средний балл.

24. В соревнованиях по десятиборью участвуют 8 спортсменов. Известны результаты в баллах по каждому из видов. Вывести фамилии спортсменов и их результаты в сумме в порядке занятых мест.

25. Известен курс акций 5 фирм за каждый из месяцев календарного года. За второе полугодие для каждой фирмы определить месяц максимального прироста курса. Если таких несколько, то вывести последний из них.

Задание ИБ.3. Задачи на сортировку и упорядочивание элементов массива.

1. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом. Выполнить сортировку по возрастанию между максимальным и первым элементом.

2. Дан массив из 25 вещественных чисел. Записать в этот же массив сначала все положительные числа и нули, а затем все отрицательные, сохраняя порядок их следования.

3. В массиве из 10 целых чисел найти наименьший элемент. Выполнить сортировку по невозрастанию между минимальным и последним элементом.

4. В массиве из 15 вещественных чисел найти наибольший элемент. Выполнить сортировку по неубыванию между максимальным и последним элементом.

5. В массиве из 25 вещественных чисел найти наименьший элемент. Выполнить сортировку по убыванию между минимальным и первым элементом.

6. Упорядочить по неубыванию массив, содержащий 20 целых чисел.

7. Упорядочить по невозрастанию массив, содержащий 10 целых чисел.

8. Упорядочить по неубыванию массив, содержащий 15 вещественных чисел.

9. Упорядочить по невозрастанию массив, содержащий 25 вещественных чисел.

10. Дан массив целых чисел, содержащий 20 элементов, записать в этот же массив сначала все отрицательные числа и нули, затем все положительные, сохраняя порядок их следования.

11. Дан массив целых чисел, содержащий 10 элементов, записать в этот же массив сначала все положительные числа, а затем все отрицательные и нули, сохраняя порядок их следования.

12. Дан массив вещественных чисел, содержащий 15 элементов, записать в этот же массив сначала все отрицательные числа, а затем все положительные и нули, сохраняя порядок их следования.

13. Дан двумерный массив, содержащий 5 строк и 3 столбца. Элементами массива являются целые числа. Переставить строки таким образом, чтобы элементы первого столбца были упорядочены по неубыванию.

14. Дан двумерный массив, содержащий 4 строки и 5 столбцов. Элементами массива являются целые числа. Переставить строки таким образом, чтобы элементы пятого столбца были упорядочены по невозрастанию.

15. Дан двумерный массив, содержащий 3 строки и 4 столбца. Элементами массива являются вещественные числа. Переставить строки таким образом, чтобы элементы первого столбца были упорядочены по невозрастанию.

16. Дан двумерный массив, содержащий 4 строки и 4 столбца. Элементами массива являются вещественные числа. Переставить строки таким образом, чтобы элементы второго столбца были упорядочены по невозрастанию.

17. Дан двумерный массив, содержащий 2 строки и 4 столбца. Элементами массива являются целые числа. Переставить столбцы таким образом, чтобы элементы первой строки были упорядочены по невозрастанию.

18. Дан двумерный массив, содержащий 3 строки и 4 столбца. Элементами массива являются целые числа. Переставить столбцы таким образом, чтобы элементы второй строки были упорядочены по невозрастанию.

19. Дан двумерный массив, содержащий 3 строки и 2 столбца. Элементами массива являются целые числа. Переставить столбцы таким образом, чтобы элементы третьей строки были упорядочены по невозрастанию.

20. Дан массив целых чисел, содержащий 15 элементов, записать в этот же массив сначала все отрицательные числа, а затем все единицы, все положительные и нули, сохраняя порядок их следования.

21. Дан одномерный массив $A[n]$. Упорядочить его по неубыванию до первого максимального числа, а от него до конца массива по невозрастанию.

22. Определите номер наибольшего элемента массива A и наибольшего значения среди модулей элементов массива A . Упорядочить массив по убыванию абсолютных величин.

Лабораторная работа № 7

СТРУКТУРЫ

Массивы позволяют обращаться с набором логически связанных однотипных элементов как с единым целым. Если же требуется хранить набор разнородных, но логически связанных данных, описывающих, например, состояние некоторого объекта реального мира, используются *структуры*. Структуры могут быть простые и связанные указателями в цепь, дерево или сеть.

Структура состоит из элементов различных типов, причем каждый элемент структуры имеет собственное имя. Описание структуры имеет вид

```
struct [имя_типа]
{
    [список описаний]
}[список_переменных];
```

Список описаний содержит типы и имена элементов структуры.

Например, структура *stud*

```
struct stud
{
    char name[20];           /*Фамилия и инициалы*/
    struct
    {
        int day;             /*День*/
        char month[9];       /*Месяц*/
        int year;            /*Год рождения*/
    } date;
    char address[40]; /*Адрес*/
} student;
```

состоит из трех элементов (одномерного символьного массива *name[20]*, структуры *date* и еще одного символьного массива *address[40]*). Второй элемент, в свою очередь, структура *date* состоит также из трех элементов (переменной целого типа *day*, одномерного символьного массива *month[9]* и переменной целого типа *year*). Таким образом, структура *stud* содержит фамилию, дату рождения и адрес местожительства студента, т.е. структура группирует различные данные, относящиеся к конкретному человеку.

Для доступа к отдельным элементам структуры имеются две операции: точка и стрелка, за которыми следует имя элемента. Какую из них следует применять, зависит от того, имеете ли вы дело с самой переменной-структурой или у вас есть только указатель на нее. С именем переменной применяется точка, с указателем – стрелка. Доступ к отдельному элементу структуры осуществляется по составному имени, образованному из имени самой структуры и имени элемента, разделенных точкой. Например, *student.name* – фамилия студента, *student.date.year* – его год рождения. Имея в виду предыдущее определение, можно было бы написать:

```
stud *p,s;
p->name="Ivanov A.A.";
```

Или

```
s.name="Ivanov A.A.";
```

Задача 7.1. Найти сумму, разность и произведение двух комплексных чисел.

```
#include <stdio.h>
#include <iostream.h>
// Описание структуры
struct
{
    int re, // Действительная часть комплексного числа
        im; // Коэффициент при мнимой части числа
}a,b,c,d,e;

int main()
{
    cout<<"Enter complex a (re and im): ";
    cin>>a.re>>a.im;
    cout<<"Enter complex b (re and im): ";
    cin>>b.re>>b.im;
    c.re=a.re+b.re;
    c.im=a.im+b.im;
    d.re=a.re-b.re;
    d.im=a.im-b.im;
    e.re=a.re*b.re+(-1)*(a.im*b.im);
    e.im=a.re*b.im+a.im*b.re;
    cout<<"a="<<a.re<<"+"<<a.im<<"i)"<<endl;
    cout<<"b="<<b.re<<"+"<<b.im<<"i)"<<endl;
    cout<<"Result (c=a+b,d=a-b,e=a*b):"<<endl;
    cout<<"c="<<c.re<<"+"<<c.im<<"i)"<<endl;
    cout<<"d="<<d.re<<"+"<<d.im<<"i)"<<endl;
    cout<<"e="<<e.re<<"+"<<e.im<<"i)"<<endl;
    getchar();getchar();
    return 0;
}
```

Результат работы программы приведен на рис. 7.1.

```
Enter complex a (re and im): 2 3
Enter complex b (re and im): 5 -7
a=2+(3i)
b=5+(-7i)
Result (c=a+b,d=a-b,e=a*b):
c=7+(-4i)
d=-3+(10i)
e=31+(1i)
```

Рис. 7.1

Задача 7.2. Вывести данные по студентам, у которых есть хотя бы одна оценка «удовлетворительно».

```

#include <stdio.h>
#include <iostream.h>
#include <iomanip.h>

const int n=25,           //Максимальное кол-во студентов в
                          //группе
        m=20;           //Максимальное кол-во символов в
                          //имени и фамилии студентов
//Описание структурированного типа данных - student
struct student
{
    char    surname [m], //Фамилия
           name[m];      //Имя
    unsigned int day,     //День
           month,         //Месяц
           year,          //Год
           ball[3];       //Оценки по трем предметам
};

int main()
{
    student stud[n];      //Массив данных типа student
    cout<<"Enter count student: ";
    int k,kol=0;          //k-кол-во студентов в группе
    cin>>k;
    //Ввод информации о k-студентах группы
    for(int i=0;i<k;i++)
    {
        cout<<"Enter information about "<<i+1<<" - student"<<endl;
        cout<<"Surname: ";
        cin>>stud[i].surname;
        cout<<"Name: ";
        cin>>stud[i].name;
        cout<<"Birthday (day, month, year) ";
        cin>>stud[i].day>>stud[i].month>>stud[i].year;
        cout<<"Ball (fisica, mathematica, information) ";
        cin>>stud[i].ball[0]>>stud[i].ball[1]>>stud[i].ball[2];
    }
    cout<<endl;
    //Вывод заголовка таблицы
    cout<<setw(20)<<"Surname"<<setw(20)<<"Name"<<setw(18)<<"Birthday"<<setw(20)<<"Ball(fis,math,inf)"<<endl;
    //Вывод информации о всех студентах группы
    for(int i=0;i<k;i++)
    {
        //Подсчет кол-ва студентов, имеющих 3 хотя бы по одному
        //предмету
    }
}

```

```

        if (stud[i].ball[0]==3||stud[i].ball[1]==3||
        stud[i].ball[2]==3) kol++;
        cout<<setw(20)<<stud[i].surname<<setw(20)<<stud[i].name<<
setw(8)<<" "<<setw(2)<<setfill('0')<<stud[i].day<<'/'
        <<setw(2)<<setfill('0')<<stud[i].month<<'/'<<stud[i].year<<
setfill(' ')<<setw(16)<<stud[i].ball[0]<<' '<<stud[i].ball[1]<<'
'<<stud[i].ball[2]<<endl;
    }
    cout<<endl<<endl;
    cout<<"Result:"<<endl;
    //Если в группе есть такие студенты, то выводим
    //информацию о них
    if (kol>0)
    {
        cout<<setw(20)<<"Surname"<<setw(20)<<"Name"<<setw(18)<<"Birthday"<<setw(20)<<"Ball(fis,math,inf)"<<endl;
        for(int i=0;i<k;i++)
            if (stud[i].ball[0]==3||stud[i].ball[1]==3||stud[i].ball[2]==3)
                cout<<setw(20)<<stud[i].surname<<setw(20)<<stud[i].name<<
setw(8)<<" "<<setw(2)<<setfill('0')<<stud[i].day<<'/'
                <<setw(2)<<setfill('0')<<stud[i].month<<'/'<<stud[i].year<<setfill(' ')<<
setw(16)<<stud[i].ball[0]<<' '<<stud[i].ball[1]<<' '<<stud[i].ball[2]<<endl;
    }
    //Иначе выводим информацию, что таких студентов нет
    else cout<<"No student";
    getch();getchar();
    return 0;
}

```

Результат работы программы приведен на рис. 7.2.

```

C:\Documents and Settings\1\Рабочий стол\Задачи С++\СТРУКТУРА\Project.exe
Enter count student: 3
Enter information about 1 - student
Surname: Иванов
Name: Иван
Birthday (day, month, year) 1 2 1990
Ball (fisica, mathematica, information) 2 3 4
Enter information about 2 - student
Surname: Петров
Name: Петр
Birthday (day, month, year) 23 12 1989
Ball (fisica, mathematica, information) 5 5 5
Enter information about 3 - student
Surname: Сидоров
Name: Максим
Birthday (day, month, year) 5 5 1988
Ball (fisica, mathematica, information) 4 5 3

Surname          Name          Birthday    Ball(fis,math,inf)
Иванов           Иван          01/02/1990   2 3 4
Петров           Петр          23/12/1989   5 5 5
Сидоров          Максим        05/05/1988   4 5 3

Result:
Surname          Name          Birthday    Ball(fis,math,inf)
Иванов           Иван          01/02/1990   2 3 4
Сидоров          Максим        05/05/1988   4 5 3

```

Рис. 7.2

Присваивание будет осуществляться только в том случае, когда две структуры однотипны.

Сравнивать непосредственно две структуры нельзя. Сравнивать можно только члены этих структур.

Члены структур хранятся в памяти подряд, друг за другом. В объединении все члены перекрывают друг друга, располагаясь по одному и тому же адресу. Это означает, что одна и та же оперативная память используется для хранения значений различных переменных одинаковых или разных типов. Это целесообразно использовать в том случае, когда значения переменных объединения не используются в программе одновременно, а поочередно. Объединения используют также для преобразования типов. При этом одно и то же значение рассматривается то, как значение одного типа, то, как значение другого типа. Форма объявления объединения аналогична форме объявления структур, но вместо ключевого слова *struct* используется ключевое слово *union*, Например:

```
union sm {  
    int ival;  
    float fval;  
    char cval; } uval;
```

где *sm* – имя типа объединения; *uval* – имя переменной типа объединения; *ival*, *fval*, *cval* – переменные различных типов, занимающие одну и ту же оперативную память, выделенную переменной *uval*.

Размер оперативной памяти для размещения объединения определяется размером наименьшего поля, достаточного для размещения любой переменной объединения. В примере это четыре байта, необходимых для размещения значения типа *float*; для значений типа *int* этот объем оперативной памяти будет достаточен. Программист должен сам следить за тем, что же находится в этой памяти.

Форма обращения к элементам объединения идентична форме обращения к элементам структур. Например, в программе могут быть операторы присваивания:

```
uval.ival = 5;  
uval.fval = 3.5;  
uval.cval = 'a';
```

Имена переменных типа объединения нельзя использовать в качестве параметров функций и передавать в качестве результатов работы функций.

Объединения могут быть в составе структур. В составе объединения могут быть массивы. Комбинация массивов и структур позволяет создавать мощные структуры данных, которые могут помочь организовать информацию почти без всяких ограничений.

**Задание для индивидуального выполнения
по теме «Работа со структурами» [1]**

Задание И6. Составить список учебной группы, включающий десять человек. Для каждого студента указать: фамилию и имя, дату рождения (год, месяц и число) оценки за сессию (от трех до пяти экзаменов). Информацию о каждом студенте оформить в виде структуры, а совокупность структур объединить в массив. Составить программу, которая обеспечивает ввод полученной информации, ее просмотр в виде таблицы, а также вывод информации на экран монитора согласно конкретному варианту. В случае, если в группе нет студентов с требуемыми данными, выдать соответствующее сообщение.

1. Вывести анкетные данные студентов отличников.
2. Вывести анкетные данные всех студентов, у которых по всем предметам «удовлетворительно».
3. Вывести анкетные данные всех студентов, успевающих на «хорошо» и «отлично».
4. Вывести анкетные данные студентов, получивших по предмету физика оценку «неудовлетворительно».
5. Вывести анкетные данные студентов, получивших по предмету физика оценку «отлично».
6. Вывести анкетные данные студентов, получивших по одному из предметов оценку «хорошо», а по всем другим – «отлично».
7. Вывести список студентов, фамилии которых начинаются с буквы "А" и их оценки по всем предметам.
8. Вывести список студентов, фамилии которых начинаются с буквы "В" и их даты рождения.
9. Вывести оценки по всем предметам студентов, фамилии которых начинаются с букв "В" и "D".
10. Вывести фамилии и даты рождения студентов, не получивших ни одной оценки «удовлетворительно» по всем предметам.
11. Упорядочить список студентов по среднему баллу и вывести весь список.
12. Упорядочить список студентов по предмету физика и вывести весь список.
13. Вычислить средний балл группы и вывести список студентов, имеющих средний балл выше, чем в целом по группе (по всем предметам).
14. Вычислить средний балл группы и вывести список студентов, имеющих средний балл ниже, чем в целом по группе (по всем предметам).
15. Вычислить средний балл группы и вывести список студентов, имеющих средний балл, близкий к среднему баллу группы (отличающийся не более чем на K процентов).

16. Упорядочить список студентов по дате рождения и вывести весь список.

17. Вывести список студентов, упорядоченный по алфавиту.

18. Вывести список студентов, упорядоченный по месяцу рождения.

19. Вывести список студентов, упорядоченный по именам.

20. Вывести анкетные данные студентов, получивших по предметам физика и химия оценку «неудовлетворительно».

21. Вывести анкетные данные студентов, получивших по предмету информатика оценку «отлично».

22. Вывести анкетные данные студентов, имеющих по предметам физика и химия оценку «хорошо» или «отлично».

23. Вывести оценки студентов, фамилии которых начинаются на буквы "В", "D", "Е", по всем предметам.

24. Вывести оценки студентов, фамилии которых начинаются на буквы "В", "D", "Е", по всем предметам, имеющих средний балл по всем предметам выше, чем средний балл в целом по группе.

Лабораторная работа № 8 СТРОКИ

Строка – это последовательность символов, заканчивающаяся *символом конца строки* '\0'.

Список основных функций для работы со строками приведен в табл. 8.1.

Таблица 8.1

Основные функции для работы со строками

<i>Прототип</i>	<i>Действия</i>
int strcmp(const char *s1, const char *s2)	Добавляет строку src в конец строки dest
int strcmp(const char *s1, const char *s2)	Сравнение двух строк. При $s1 < s2$ функция возвращает отрицательное значение, $s1 = s2$ – нулевое, $s1 > s2$ – положительное
char *strcpy(char *dest, const char *src);	Копирует src в dest
size_t strlen(const char *s);	Вычисляет длину строки
char *strlwr(char *s);	Преобразует строку в нижний регистр
char *strrev(char *s);	Инвертирует строку (переворачивает)
char *strupr(char *s);	Преобразует строку в верхний регистр
double atof(const char *s);	Преобразует строку в тип float
int atoi(const char *s);	Преобразует строку в тип int
long atol(const char *s);	Преобразует строку в тип long
char *ecvt(double value, int ndig, int *dec, int *sign);	Преобразует строку в тип double
char *gcvt(double value, int ndec, char *buf);	Преобразовать тип double в строку
char *itoa(int value, char *string, int radix);	Преобразует тип int в строку
char *ltoa(long value, char *string, int radix);	Преобразует тип long в строку

Задача 8.1. Пользователем вводится строка, найти и вывести самое длинное слово.

```
#include <iostream.h>
#include <string.h>
int main()
{
    char str[1000];
    //curlen - текущая длина слова
    //maxlen - максимальная длина слова
    //index – индекс начала слова
```

```

int curlen = 0, maxlen = 0, index = 0, i;
cout << " Input line: ";
cin.get(str, 1000);
strcat(str, " ");
for(i=0; i<strlen(str)-1; i++)
{
    if (str[i]==' ' && str[i+1]!=' ')
    { //Начало слова
        index = i+1; continue;
    }
    if(str[i]!=' ' && str[i+1]==' ')
    { //Конец слова
        curlen=i-index+1;
        if(curlen>maxlen)
            maxlen = curlen;
    }
}
cout << maxlen<<endl;
cin.get();cin.get();
return 0;
}

```

Задание для индивидуального выполнения по теме «Работа со строками»

Задание И8.

1. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке последние слова всех предложений. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

2. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке первые слова всех предложений. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

3. Дан текст, состоящий из двух строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке слова, присутствующие в обеих строках одновременно. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

4. Дан текст, состоящий из двух строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке те слова, которые не присутствуют в обеих строках одновременно. Считать, что

текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

5. Дан текст, состоящий из трех строк с максимальной длиной 80 символов. Необходимо вывести все предложения данного текста в порядке неубывания их длины. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения предложений из текста создать пользовательскую функцию.

6. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке те слова, где количество гласных превышает количество согласных. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

7. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке те слова, где количество согласных превышает количество гласных. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

8. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке те слова, длина которых превышает K символов. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

9. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке те слова, длина которых не превышает K символов. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

10. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке те слова, последняя буква которых является гласной. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов

по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

11. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке те слова, последняя буква которых является согласной. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

12. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке те слова, первая буква которых является согласной, а последняя гласной. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

13. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке предпоследние слова всех предложений. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует, минимальная длина предложений — два слова. Для выделения слов из строки создать пользовательскую функцию.

14. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке вторые слова всех предложений. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует, минимальная длина предложений — два слова. Для выделения слов из строки создать пользовательскую функцию.

15. Дан текст, состоящий из трех строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке те слова, которые присутствуют в третьей строке и не присутствуют в первых двух. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

16. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в алфавитном порядке слова, начинающиеся с прописных букв. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки создать пользовательскую функцию.

17. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести в обратном алфавитном порядке слова, являющиеся палиндромами. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки и определения является ли они палиндромами, создать пользовательские функции.

18. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести все предложения данного текста, являющиеся палиндромами, в порядке невозрастания их длины. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения предложений из строки и определения является ли они палиндромами, создать пользовательские функции.

19. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести этот текст, зашифрованный кодом Цезаря. (Код Цезаря заменяет одну букву другой, отстоящей от нее на заданное количество позиций в алфавите. Например, при сдвиге, равном 1, буква А заменяется на Б, Б — на В, ..., Я — на А.) Размер сдвига символов принять равным номеру буквы в слове. Например, слово ДОМ шифруется как ЕРП. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки и зашифровки слова создать пользовательские функции.

20. Составить программу для расшифровки текста, зашифрованного в пункте 19. Для выделения слов из строки и расшифровки слова создать пользовательские функции.

21. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести этот текст, зашифрованный кодом Цезаря. (Код Цезаря заменяет одну букву другой, отстоящей от нее на заданное количество позиций в алфавите. Например, при сдвиге, равном 1, буква А заменяется на Б, Б — на В, ..., Я — на А.) Размер сдвига символов принять равным номеру буквы в предложении. Например, если предложение начинается со слова ДОМ, то оно шифруется как ЕРП. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения предложений из текста и зашифровки предложения создать пользовательские функции.

22. Составить программу для расшифровки текста, зашифрованного в пункте 21. Для выделения предложений из текста и расшифровки предложения создать пользовательские функции.

23. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести этот текст, зашифрованный кодом Цезаря. (Код Цезаря заменяет одну букву другой, отстоящей от нее на заданное количество позиций в алфавите. Например, при сдвиге, равном 1, буква А заменяется на Б, Б — на В, ..., Я — на А.) Размер сдвига символов принять равным остатку от деления длины слова на номер буквы в слове плюс единица. Например, слово ДОМ шифруется как ЕРН. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения слов из строки и зашифровки слова создать пользовательские функции.

24. Составить программу для расшифровки текста, зашифрованного в пункте 23. Для выделения слов из строки и расшифровки слова создать пользовательские функции.

25. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести этот текст, зашифрованный кодом Цезаря. (Код Цезаря заменяет одну букву другой, отстоящей от нее на заданное количество позиций в алфавите. Например, при сдвиге, равном 1, буква А заменяется на Б, Б — на В, ..., Я — на А.) Размер сдвига символов принять равным остатку от деления количества слов в предложении на номер буквы в слове плюс единица. Например, если в предложении 8 слов, слово ДОМ шифруется как ЕПР. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует. Для выделения предложений из текста и зашифровки предложения создать пользовательские функции.

26. Составить программу для расшифровки текста, зашифрованного в пункте 25. Для выделения предложений из текста и расшифровки предложения создать пользовательские функции.

27. Дан текст, состоящий из N ($2 \leq N \leq 10$) строк с максимальной длиной 80 символов. Необходимо вывести этот текст, зашифрованный кодом Цезаря. (Код Цезаря заменяет одну букву другой, отстоящей от нее на заданное количество позиций в алфавите. Например, при сдвиге, равном 1, буква А заменяется на Б, Б — на В, ..., Я — на А.) Размер сдвига символов принять равным сумме сдвигов двух предыдущих букв. Для первой буквы принять сдвиг равным 0, а для второй — 1. Например, слово КОШКА шифруется как КПЦМГ. Считать, что текст написан синтаксически грамотно, в качестве знаков препинания используются точка и запятая, слова состоят только из букв, перенос слов по слогам отсутствует.

28. Составить программу для расшифровки текста, зашифрованного в пункте 27. Для выделения слов из строки и расшифровки слова создать пользовательские функции.

Лабораторная работа № 9

ФУНКЦИИ

Мощность языка C/C++ во многом определяется легкостью и гибкостью в определении и использовании функций в C/C++-программах. В отличие от других языков программирования высокого уровня в языке C/C++ нет деления на процедуры, подпрограммы и функции, здесь вся программа строится только из функций.

Функция – это совокупность объявлений и операторов, обычно предназначенная для решения определенной задачи. Каждая функция должна иметь имя, которое используется для ее объявления, определения и вызова. В любой программе на СИ должна быть функция с именем *main* (главная функция), именно с этой функции, в каком бы месте программы она не находилась, начинается выполнение программы.

При вызове функции ей при помощи аргументов (формальных параметров) могут быть переданы некоторые значения (фактические параметры), используемые во время выполнения функции. Функция может возвращать некоторое (одно!) значение. Это возвращаемое значение и есть результат выполнения функции, который при выполнении программы подставляется в точку вызова функции, где бы этот вызов ни встретился. Допускается также использовать функции, не имеющие аргументов, и функции, не возвращающие никаких значений. Действие таких функций может состоять, например, в изменении значений некоторых переменных, выводе на печать некоторых текстов и т.п.

С использованием функций в языке C/C++ связаны три понятия – определение функции (описание действий, выполняемых функцией), объявление функции (здание формы обращения к функции) и вызов функции.

Определение функции задает тип возвращаемого значения, имя функции, типы и число формальных параметров, а также объявления переменных и операторы, называемые телом функции и определяющие действие функции. В определении функции также может быть задан класс памяти.

Пример:

```
int rus (unsigned char r)
{ if (r>='A')
    return 1;
  else
    return 0;
}
```

В данном примере определена функция с именем *rus*, имеющая один параметр с именем *r* и типом *unsigned char*. Функция возвращает целое значение, равное 1, если параметр функции является буквой русского алфавита, или 0 в противном случае.

Время жизни и область видимости программных объектов
Время жизни переменной (глобальной или локальной) определяется по следующим правилам.

1. Переменная, объявленная глобально (т.е. вне всех блоков), существует на протяжении всего времени выполнения программы.

2. Локальные переменные (т.е. объявленные внутри блока) с классом памяти *register* или *auto* имеют время жизни только на период выполнения того блока, в котором они объявлены. Если локальная переменная объявлена с классом памяти *static* или *extern*, то она имеет время жизни на период выполнения всей программы.

Видимость переменных и функций в программе определяется следующими правилами.

1. Переменная, объявленная или определенная глобально, видима от точки объявления или определения до конца исходного файла. Можно сделать переменную видимой и в других исходных файлах, для чего в этих файлах следует ее объявить с классом памяти *extern*.

2. Переменная, объявленная или определенная локально, видима от точки объявления или определения до конца текущего блока. Такая переменная называется локальной.

3. Переменные из объемлющих блоков, включая переменные, объявленные на глобальном уровне, видимы во внутренних блоках. Эту видимость называют вложенной. Если переменная, объявленная внутри блока, имеет то же имя, что и переменная, объявленная в объемлющем блоке, то это разные переменные, и переменная из объемлющего блока во внутреннем блоке будет невидимой.

4. Функции с классом памяти *static* видимы только в исходном файле, в котором они определены. Всякие другие функции видимы во всей программе.

Имена формальных параметров, объявленные в списке параметров прототипа функции, видимы только от точки объявления параметра до конца объявления функции.

Для некоторых инструкций языка C++ разрешено объявлять переменные внутри управляющей части. Например, в цикле *for* переменную можно определить внутри инструкции инициализации. Подобные переменные видны только в локальной области самого цикла *for* и вложенных в него (это верно для стандарта C++, в предыдущих версиях языка поведение было иным). Аналогично переменная может быть объявлена внутри условия инструкций *if* и *switch*, а также внутри условия циклов *while* и *for*.

```
if ( int *pi = getValue() )
{ // pi != 0 -- *pi можно использовать здесь
}
else
{ // здесь pi тоже видна
}
```

Задача 9.1. Дан одномерный массив из 100 элементов, состоящий из случайных вещественных чисел в диапазоне от 0 до 100. Определить минимальный, максимальный элемент и среднее значение. Ввод/вывод элементов массива и поиск оформить в виде функции.

Листинг программы

```
#include <iostream.h>
```

```
#include <iomanip.h>
```

```
void init(float[], int);
```

```
float sred(float*, int);
```

```
float *maxmin(float*, int, float&);
```

```
void print(float[], int);
```

```
int main()
```

```
{
```

```
    randomize();
```

```
    const int n=100;
```

```
    float mas[n],min;
```

```
    init(mas,n);
```

```
    cout<<"Data: "<<endl;
```

```
    print(mas,n);
```

```
    cout<<endl<<"Result: "<<endl;
```

```
    cout<<"sred = "<<sred(mas,n)<<" max = "
```

```
        <<*maxmin(mas,n,min);
```

```
    cout<<" min = "<<min<<endl;
```

```
    getchar();
```

```
    return 0;
```

```
}
```

```
void init(float a[],int k)
```

```
{
```

```
    for (int i=0;i<k;i++)
```

```
        a[i]=random(100)+(random(99)+1)/100.;
```

```
}
```

```
void print(float a[],int k)
```

```
{
```

```
    for (int i=0;i<k;i++)
```

```
        (i+1)%10==0 ? (cout<<setw(6)<<a[i]<<endl) :
```

```
        (cout<<setw(6)<<a[i]);
```

```
}
```

```
float sred(float*a,int k)
```

```
{
```

```
    float sum=0;
```

```
    for (int i=0;i<k;i++)
```

```
        sum+=a[i];
```

```

    return sum/k;
}

float *maxmin(float *a, int k, float &min)
{
    float max=a[0];
    min=a[0];
    for (int i=1;i<k;i++)
    {
        min=a[i]<min?a[i]:min;
        max=a[i]>max?a[i]:max;
    }
    return &max;
}

```

Результат работы программы задачи 9.1 представлен на рис.9.1.

```

Data:
70.11 84.89 35.17 68.35 51.82 18.98 81.3 63.76 14.37 61.71
42.63 25.07 58.3 72.51 32.82 1.07 73.18 39.81 37.39 9.96
98.58 39.53 36.53 72.08 36.48 96.26 44.56 53.18 61.62 62.26
44.55 42.19 50.04 46.38 90.3 93.31 68.52 56.91 67.81 13.99
10.05 95.51 61.53 50.28 16.37 0.36 86.2 63.75 45.38 62.41
21.61 19.56 86.45 77.31 89.32 78.39 38.96 41.2 78.1 68.05
70.8 4.54 18.21 91.96 4.81 26.72 86.43 71.58 49.44 44.38
11.2 91.26 67.73 48.8 22.37 74.08 10.3 22.66 73.6 68.14
44.76 92.66 56.97 85.39 69.87 37.14 80.84 42.75 27.15 73.99
92.7 7.35 55.89 39.44 0.11 4.2 13.05 5.72 55.08 86.35

Result:
sred = 51.3969 max = 98.58 min = 0.11

```

Рис. 9.1

Задача 9.2. Спортсмены на соревнованиях совершают 4 попытки в тройных прыжках в длину. Определить средний результат для каждого участника и спортсмена, показавшего наилучший результат. Нахождение спортсмена, показавшего наилучший результат, оформить в виде функции.

Листинг программы

```

#include<iostream.h>
#include<omanip.h>
void init(float**,int,int);
int poisk(float**,int,int,float&);
void print(float**,int,int);

int main()
{
    randomize();
    int n, m, num;
    cout<<"Enter count sportsman: ";
    cin>>n;

```

```

float bestrez=0,
    **rez=(float**)calloc(n, sizeof(float*));
cout<<"Enter count jump: ";
cin>>m;
for(int i=0;i<n;i++)
    rez[i]=(float*)calloc(m+1,sizeof(float));
init(rez,n,m);
num=poisk(rez,n,m,bestrez);
print(rez,n,m);
cout<<"The best jump "<<num<<" - sportsman " << bestrez
    <<" m.!",
for(int i=0;i<n;i++)
    free(rez[i]);
free(rez);
getchar();getchar();
return 0;
}

```

```

void init(float **a,int k,int l)
{
    for(int i=0;i<k;i++)
        for(int j=0;j<l;j++)
            a[i][j]=random(4)+13+(random(99)+1)/100.;
}

```

```

int poisk(float **a, int k, int l, float &bestrez)
{
    int nn=0;
    for(int i=0;i<k;i++)
    {
        a[i][l]=0;
        for(int j=0;j<l;j++)
        {
            a[i][l]+=a[i][j];
            if (a[i][l]>bestrez)
            {
                bestrez=a[i][l];
                nn=i+1;
            }
        }
        a[i][l]/=l;
    }
    return nn;
}

```

```

void print(float **a,int k,int l)
{

```

```

cout<<"Result: "<<endl;
for(int i=0;i<k;i++)
{
    cout<<"Sportsman "<<setw(4)<<(i+1)<<": ";
    for(int j=0;j<l+1;j++)
        j%l?(cout<<setw(6)<<a[i][j]):(cout<<setw(6) <<
            " sr="<<a[i][l]);
    cout<<endl;
}
}

```

Результат работы программы задачи 9.2 представлен на рис.9.2.

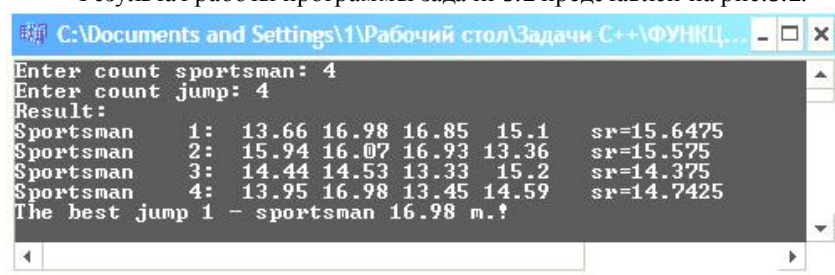


Рис. 9.2

Задание для индивидуального выполнения по теме «Работа с функциями»

Задание И.9.

1. Дано N десятков целых чисел. Определить, сколько из них могут составлять геометрическую прогрессию. Проверку оформить в виде функции.

2. Дано N десятков целых чисел. Определить, сколько из них могут составлять арифметическую прогрессию. Проверку оформить в виде функции.

3. Дано N десятков целых чисел. Определить, сколько из них могут составлять ряд Фибоначчи. Первое число Фибоначчи равно 0, второе – 1. Каждое последующее равно сумме двух предыдущих. Проверку оформить в виде функции.

4. Дано N пар чисел, представляющих собой координаты точек на плоскости. Найти R — радиус наименьшей окружности с центром в начале координат, в которую попадают все точки. Определение расстояния от точки до начала координат оформить в виде функции.

5. Известны оценки группы студентов за сессию. В группе 20 студентов, в сессии 4 экзамена. Определить суммарную стипендию. Считать, что стипендия в размере R рублей начисляется студентам, сдавшим сессию без троек, а отличники получают стипендию, повышенную на 25%. Подсчет стипендии студента оформить в виде функции.

6. Известен расход электроэнергии по всем квартирам 24-квартирного дома. Определить суммарную плату за электричество. При расходе до 100 кВт·ч на человека берется тариф R копеек за 1 кВт·ч, в случае превышения нормы тариф возрастает на 20%. Подсчет платы для квартиры оформить в виде функции.

7. Известна ежемесячная заработная плата персонала предприятия в течение календарного года. Вывести фамилии тех сотрудников, у которых годовая заработная плата выше средней. Считать, что штат предприятия составляет 7 человек. Подсчет годовой зарплаты работника оформить в виде функции.

8. Известна ежемесячная заработная плата персонала предприятия в течение календарного года. Вывести фамилии сотрудников с минимальной и максимальной годовой заработной платой. Считать, что штат предприятия составляет 8 человек. Подсчет годовой зарплаты работника оформить в виде функции.

9. Дан одномерный массив из 100 случайных целых чисел в диапазоне от 5 до 25 включительно. Вывести все числа, которые максимально часто встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.

10. Дан одномерный массив из 150 случайных целых чисел в диапазоне от 14 до 37 включительно. Вывести те числа, которые наиболее редко встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.

11. Дан одномерный массив из 50 случайных целых чисел в диапазоне от 10 до 85 включительно. Вывести в порядке возрастания те числа из данного диапазона, которые ни разу не встречаются в массиве. Создать функцию для поиска элемента в массиве.

12. Дан одномерный массив из 40 случайных целых чисел в диапазоне от 16 до 89 включительно. Вывести минимальное и максимальное числа из данного диапазона, которые ни разу не встречаются в массиве. Создать функцию для поиска элемента в массиве.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Серебренникова И.Г., Воронина О.Б. Структурный тип данных: методические указания и варианты индивидуальных заданий по дисциплине «Программирование на языке высокого уровня» для студентов всех форм обучения специальности 220400. – Магнитогорск: МГТУ, 2003. – 21 с.
2. Экономическая информатика: учебник для вузов / под ред., проф. В.В. Евдокимова. – СПб., 1997. – 592 с.
3. Гричишин Я.Т. Алгоритмы и программы на Бейсике. – М.: Просвещение, 1988.
4. Чернов Б.И. Программирование на алгоритмических языках Бейсик, Фортран, Паскаль. – М.: Просвещение, 1991.
5. Поддубная Л.М. Мне нравится Паскаль. – М.: Радио и связь, 1992.
6. Бабушкина В.Е. и др. Практикум по программированию на Turbo Pascal. – М.: Бином, 1998.
7. Программирование на языке VISUAL BASIC FOR APPLICATION: учебное пособие/ Логунова О.С., Ильина Е.А., Кирпичева Н.Т., Вьяльцина Т.Н., Тутарова В.Д. – Магнитогорск: МГТУ, 2000.
8. Юркин А. Задачник по программированию. – СПб.: Питер, 2002. – 192 с.
9. Перельман Я.И. Живая математика. – М.: Наука, 1978. – 160 с.
10. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: практикум. – СПб.: Питер, 2005. – 239 с.
11. Есипов А.С., Паньгина Н.Н., Громада М.И. Информатика. Сборник задач и решений для общеобразовательных учебных заведений. – СПб.: Наука и Техника, 2001. – 368 с.
12. Филиппова Н.В. и др. Практикум по программированию: учеб. пособие. – Магнитогорск: МГТУ им. Г.И.Носова, 2000. – 84 с.
13. Алексеев Е.Р., Чеснокова О.В. Турбо Паскаль 7.0. – М.: НТ Пресс, 2006. – 320 с.

ПРИЛОЖЕНИЕ
Таблица П1

Знаки операций

<i>Знак операции</i>	<i>Операция</i>	<i>Группа операций</i>
*	Умножение	Мультипликативные
/	Деление	
%	Остаток от деления	
+	Сложение	Аддитивные
-	Вычитание	
<<	Сдвиг влево	Операции сдвига
>>	Сдвиг вправо	
<	Меньше	Операции отношения
<=	Меньше или равно	
>=	Больше или равно	
==	Равно	
!=	Не равно	
&	Поразрядное И	Поразрядные операции
	Поразрядное ИЛИ	
^	Поразрядное исключающее ИЛИ	
&&	Логическое И	Логические операции
	Логическое ИЛИ	
,	Последовательное вычисление	Последовательного вычисления
=	Присваивание	Операции присваивания
*=	Умножение с присваиванием	
/=	Деление с присваиванием	
%=	Остаток от деления с присваиванием	
-=	Вычитание с присваиванием	
+=	Сложение с присваиванием	
<<=	Сдвиг влево с присваиванием	
>>=	Сдвиг вправо с присваиванием	
&=	Поразрядное И с присваиванием	
=	Поразрядное ИЛИ с присваиванием	
^=	Поразрядное исключающее ИЛИ с присваиванием	

Таблица П2

Приоритеты операций и порядок вычислений

<i>Приоритет</i>	<i>Знак операции</i>	<i>Типы операции</i>	<i>Порядок выполнения</i>
1	() [] -> :: .	Выражение	Слева направо
2	- ~ ! * & ++ -- sizeof приве- дение типов	Унарные	Справа налево
3	* / %	Мультипликативные	Слева направо
4	+ -	Аддитивные	
5	<< >>	Сдвиг	
6	< > <= >=	Отношение	
7	== !=	Отношение (равенство)	
8	&	Поразрядное И	
9	^	Поразрядное исклю- чающее ИЛИ	
10		Поразрядное ИЛИ	
11	&&	Логическое И	
12		Логическое ИЛИ	
13	? :	Условная	
14	= *= /= %= += -= &= = >>= <<= ^=	Простое и составное присваивание	Справа налево
15	,	Последовательное вы- числение	Слева направо

Св.темплан 2009, поз.52

Заявки на книгу присылать по адресу:
455000, Магнитогорск, пр. Ленина, 38,
ГОУ ВПО «МГТУ», кафедра ВТиПМ
Тел.: (3519) 29-85-63; факс: 29-84-26

ТОРЧИНСКИЙ Вадим Ефимович
ТУТАРОВА Власта Дилияуровна
КАЛИТАЕВ Александр Николаевич
ИЛЬИНА Елена Александровна

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ C++

Учебное пособие

Редактор Н.В.Кутекина

Оператор компьютерной верстки Е.А.Назарова

Подписано в печать 15.06.09. Формат 60х84 1/16.

Плоская печать. Усл.печ.л. 6,75. Уч.-изд.л. 7,83.

Заказ 403.

Бумага тип.№ 1.

Тираж 100 экз.



Издательский центр ГОУ ВПО «МГТУ»
455000, Магнитогорск, пр. Ленина, 38
Полиграфический участок ГОУ ВПО «МГТУ»