

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Южно-Уральский государственный университет
Кафедра экономико-математических методов и статистики

Ч48.я7
П168

А. В. Панюков, Т. А. Панюкова

Практикум по программированию

*Учебное пособие для студентов специальностей «Статистика»
и «Математические методы в экономике»*

Челябинск
Издательство ЮУрГУ
2007

ББК Ч481.266.3.я7
П168

*Одобрено
учебно-методической комиссией факультета экономики и управления*

Рецензенты: А.Д. Липенков, А.Н. Тырсин

П168 **Панюков, А.В.**

Практикум по программированию: учебное пособие для студентов специальностей «Статистика» и «Математические методы в экономике» / А.В. Панюков, Т.А. Панюкова. – Челябинск: Изд-во ЮУрГУ, 2007. – 75 с.

Практикум содержит рекомендации по разработке алгоритмов и программ, а также варианты заданий по дисциплине «Алгоритмизация и языки программирования», читаемой студентам специальности 080116 – «Математические методы в экономике», и дисциплине «Программирование», читаемой студентам специальности 080601 – «Статистика». Набор задач используется для выполнения лабораторных работ. Практикум может быть полезен также студентам других специальностей, имеющих в своем учебном плане данную дисциплину.

ББК Ч481.266.3.я7

© Издательство ЮУрГУ, 2007.

Оглавление

Введение.....	5
Виды работ, составляющие задание	5
Порядок и сроки сдачи отдельных этапов выполнения задания	5
Требования по оформлению программы.....	6
Рекомендации по разработке алгоритма, представления данных и набора тестов..	6
Анализ задачи	6
Разработка алгоритма	7
Обоснование алгоритма	11
Средства повышения надежности алгоритма	13
Выбор и обоснование набора тестов	13
Описание разработанных алгоритмов	15
Выбор представления данных	16
Кодирование алгоритма (программирование).....	20
Варианты задач для начинающих.....	23
Общие требования	23
Вариант 1	24
Вариант 2	25
Вариант 3	26
Вариант 4	28
Вариант 5	29
Вариант 6	30
Вариант 7	32
Вариант 8	33
Вариант 9	35
Вариант 10	36
Вариант 11	37
Вариант 12	39
Вариант 13	40
Вариант 14	41
Вариант 15	42
Вариант 16	44
Вариант 17	45
Вариант 18	46
Задачи для продолжающих	48
Общие требования	48
Вариант 1	49
Вариант 2	49
Вариант 3	50
Вариант 4	50
Вариант 5	51
Вариант 6	52
Вариант 7	52

Вариант 8	53
Вариант 9	54
Вариант 10	54
Вариант 11	55
Вариант 12	55
Вариант 13	56
Вариант 14	57
Вариант 15	57
Вариант 16	58
Вариант 17	58
Вариант 18	59
Задания на курсовую работу	60
Задание 1. Общие положения	60
Задание 2	62
Дополнительные задачи (для продвинутых)	64
Геометрия	64
Матрицы, векторы	67
Последовательности, тексты, предложения и слова	71
Задачи с целыми числами	73
Библиографический список	75

Введение

Первоочередная цель, которая ставится студенту при выполнении заданий по дисциплине «Алгоритмизация и языки программирования», – это практическое освоение всех этапов разработки надежной и наглядной программы (которую принято коротко называть С++-программой) для решения несложной задачи на компьютере, начиная от анализа условия задачи и кончая оформлением и защитой отчета по построенной программе.

Важно, чтобы уже в начале семестра студент правильно понимал условие задачи и знал все требования, предъявляемые преподавателем к выполнению задания. Один из путей достижения этого – регулярное обсуждение с преподавателем понимания задания, выданного студенту.

Виды работ, составляющие задание

Задание на самостоятельную или лабораторную работу состоит из конкретного набора задач, которые должны быть решены в течение семестра на персональном компьютере с помощью системы программирования С++ Builder. Выполнение задания включает следующие традиционные для программиста виды работ:

- 1) анализ условия задачи и выработка подхода к ее решению;
- 2) пошаговая разработка (на основании выбранного подхода) алгоритма решения и его описание;
- 3) обоснование алгоритма;
- 4) выбор и обоснование представления для входных, выходных и промежуточных данных;
- 5) кодирование алгоритма, т. е. его запись на языке программирования С++;
- 6) выбор набора тестов, на которых будет проверяться программа;
- 7) отладка программы и демонстрация ее правильной работы на выбранном наборе тестов.

Это разбиение условно в том смысле, что фактически некоторые виды работ тесно переплетаются и выполнение их обычно составляет единый процесс. Например, строить набор тестов удобнее одновременно с проработкой алгоритма, а обосновывать правильность его работы – путем пошаговой демонстрации процесса его построения.

Порядок и сроки сдачи отдельных этапов выполнения задания

Задание сдается в виде рабочего и итогового отчетов, каждый из которых оценивается преподавателем. Полученные оценки равноправно влияют на итоговую оценку, выставяемую студенту по компьютерному практикуму.

Рабочий отчет сдается до середины семестра и включает анализ задачи, словесное описание алгоритма и обоснование его правильности, описание представлений данных и обоснование их выбора, тестовый набор и его обоснование, а также текст С++-программы.

В итоговый отчет дополнительно включается описание процесса отладки, а также листинги, демонстрирующие правильность работы программы на выбранном тестовом наборе.

Перечисленные требования в ходе учебного процесса могут быть изменены или уточнены преподавателем.

Требования по оформлению программы

Программа оформляется в виде набора функций, полностью решающих сформулированную задачу, и из нескольких (по числу тестов) обращений к нему.

Программа обязательно должна быть **наглядной**, т. е. хорошо структурированной, с мнемоническими (несущими смысловую нагрузку) именами переменных и т. п.

Обязательным требованием является высокая надежность разработанной программы. В ней должно содержаться достаточное число так называемых *стопоров* ошибок (утверждений и защитных распечаток). Тестовый набор, на котором демонстрируется правильность работы программы, обязан быть полным.

Представление вводимой информации, а также вид печатаемых программой результатов должны быть естественными для человека.

Рекомендации по разработке алгоритма, представления данных и набора тестов

Анализ задачи

Разработка алгоритма представляет собой задачу на построение. Поэтому, как обычно в таких случаях, необходим **этап анализа задачи** (можно, например, вспомнить о методе решения геометрических задач на построение).

Он позволяет установить:

- что именно является входом и выходом будущего алгоритма;
- выделить основные отношения между входными и выходными объектами и их компонентами;
- выделить подцели, которых нужно достичь для решения задачи;
- и как следствие предыдущих пунктов, выработать подход к построению алгоритма.

Результатом этого этапа должна быть **спецификация алгоритма**, т. е. формулировка в самом общем виде того, что (в рамках выбранного подхода) должен делать алгоритм для переработки входных данных в выходные.

В формулировках задач, как правило, отсутствуют какие-либо ограничения на размер входных данных, подлежащих обработке. Ясно, однако, что ресурсы вычислительной системы не беспредельны как в смысле объема памяти, имеющейся для хранения программы и ее данных, так и в смысле времени выполнения программы.

Поэтому спецификацию алгоритма нельзя признать полной, если не описан явно (через систему ограничений) класс тех входных данных, которые, безусловно, должны обрабатываться алгоритмом.

Например, если обрабатываются **тексты**, то описанием класса входных данных является кроме описания структуры текста еще и максимальная допустимая длина входного текста.

Если обрабатываются **матрицы**, то такое описание должно, например, включать пару чисел:

- максимально допустимое число строк;
- максимально допустимое число столбцов в матрице и т. п.

Предполагается, что при выборе ограничений студенту необходимо не сужать, а максимально расширять класс входных данных (и, таким образом, разрабатывать как можно более универсальную программу) в рамках тех вычислительных ресурсов, которые ему отводятся для выполнения задания на ЭВМ.

Разработка алгоритма

Разработка алгоритма (а также его обоснование и, если это необходимо, модификация) существенно осложняется, если разработчик не придерживается с самого начала некоторой **дисциплины**, позволяющей на каждом этапе разработки четко выделять необходимые подцели и проследивать взаимосвязь между ними.

Такой дисциплиной, получившей в последние годы широкое распространение, является **метод пошаговой разработки**.

Суть этого метода состоит в том, что **алгоритм разрабатывается «сверху вниз»**, начиная с его спецификации, полученной в результате анализа задачи. На каждом этапе принимается небольшое число решений, приводящих к постепенной детализации (уточнению) управляющей и информационной структуры алгоритма. Таким образом, получается последовательность все более детальных спецификаций алгоритма, каждая из которых все более приближается к окончательной версии программы (формульно-словесная запись алгоритма, блок-схемы, псевдокод, запись на языке программирования).

Этот подход позволяет **разбить алгоритм на части** (модули), каждая из которых решает самостоятельную (как правило, небольшую) подзадачу. В результате появляется возможность сосредоточить усилия на решении подзадачи, реализуемой в виде отдельной процедуры или функции. Более того, при таком

подходе появляется возможность распределения решения задачи между отдельными исполнителями группы разработчиков.

Связи по управлению между модулями осуществляются посредством соответствующих обращений к ним (вызовов), а передача информации от одного модуля к другому производится через параметры и глобальные переменные.

Рассмотрим применение метода пошаговой разработки на небольшом примере.

Пример. Пусть по заданному целому числу X требуется напечатать матрицу A размером 10×10 следующим образом:

$$\begin{array}{cccc} 1 & X & X^2 & X^9 \\ X & X^2 & X^3 & X^{10} \\ & & \ddots & \\ X^9 & X^{10} & X^{11} & X^{18} \end{array}$$

Заметим, что каждая последующая строка матрицы получена из предыдущей ее умножением на X .

Общую спецификацию решения этой задачи можно записать в виде, изображенном на рис. 1.

```
#define n 10          // Размер матрицы
#include <iostream.h>  // Консольный ввод/вывод
double A[n][n];      // Статическое выделение памяти
void Calculate_A(double );
void Write_A(void);
int main () {
    double x;
    cout << "Программа напечатает матрицу
    {x^(i+j)}, i,j=0,1,...,10" << '\n' << "Введите x ";
    cin >> x;
    Calculate_A(x);    // Вычисление элементов матрицы
    Write_A();         // Печать матрицы
    return(0);
}
```

Рис. 1. Текст основной программы

В тексте программы присутствуют две *подзадачи* (решаемые функциями `Calculate_A()` и `Write_A()`), сформулированные в виде нераскрытых шагов алгоритма.

Переход от этой спецификации к более детальной осуществляется с помощью функции `Calculate_A()`, вычисляющей все элементы матрицы A , и функции `Write_A()`, предназначенной для печати данной матрицы. Заметим, что печать матрицы осуществляется после вычисления *всех элементов матрицы*.

Однако можно было предусмотреть и другую детализацию программы, а именно: выдавать на печать *текущую строку матрицы* сразу же после вычисления *всех элементов строки матрицы*. Более того, можно показать, что такая детализация в конце концов приводит к более эффективной программе, нежели приведенная выше.

Остановим выбор на спецификации, показанной на рис. 1. Реализация функции `Calculate_A()` определяется выбранным способом заполнения элементов матрицы `A`. Например, естественно заполнять матрицу `A` по строкам. Заполним первую строку (с помощью функции `FirstLine()`).

Чтобы избежать повторного вычисления совпадающих элементов, имеющих место в последующих строках, поступим следующим образом. На первые места в каждой следующей строке поместим девять элементов предыдущей, а последний элемент – вычислим, умножив предыдущий на `X` (приведенная процедура осуществляется функцией `NextLine()`). С ориентацией на этот подход получаем детализацию функции `Calculate_A()`, приведенную на рис. 2.

```
void FirstLine(double );
void NextLine(double, int);
void Calculate_A(double x){
    FirstLine(x); // Заполнить первую строку матрицы A
    for (int i=0; i<n; i++)NextLine(x,i); // Запол-
        нить (i+1)-ю строку
    };
```

Рис. 2. Текст функции `Calculate_A()`

Дальнейшая детализация описаний функций программы приведена на рис. 3.

```
void FirstLine(double x){
    A[0][0]=1;
    for (int i=1; i<n; i++) A[0][i]= A[0][i-1]*x;
    };
void NextLine(double x, int i){
    K=n-1;
    for (int j=0; j<K; j++) A[i][j]= A[i-1][j+1];
    A[i][K]=A[i-1][K]*x;
    };
```

Рис. 3. Текст функций `FirstLine()` и `NextLine()`

Итак, мы видим, как постепенная детализация приводит к появлению в спецификации алгоритма конструкций языка программирования (в данном случае конструкций конкретного языка программирования `C++` вместо абстрактно сформулированных шагов).

Функцию вывода `Write_A()` запишем в виде, приведенном на рис. 4.

Заметим, что подзадачи, сформулированные в виде нераскрытых шагов алгоритма, приведенного на рис. 1, решаются независимо друг от друга.

Объединение фрагментов, приведенных на рис. 1 – 4 дает полный текст программы. Причем, замечательно то, что порядок функций в общей записи программы может быть произвольным (единственное, что следует учитывать: функция должна приводиться ниже ее описания).

```
void Write_A() {  
    cout<<' \n'<<"Матрица A"<<' \n';  
    for (int i=0; i<n; i++ ) {  
        for (int j=0; j<n; j++) cout<<A[i][j]<<' \t';  
        cout<<' \n';  
    }  
};
```

Рис. 4. Функция вывода `Write_A()`

Наряду с использованием метода пошаговой разработки необходимо также иметь в виду следующие **факторы**, которые могут существенно повлиять на разрабатываемый алгоритм.

1. **Средства, предоставляемые** тем **языком**, на котором алгоритм будет запрограммирован. Например, в языке C++ допускаются так называемые рекурсивные процедуры (функции, определяемые через самих себя), в то время как в некоторых других языках их нет. Таким образом, при использовании различных языков имеется возможность разрабатывать существенно различные алгоритмы.

2. **Структуры данных**, на которые ориентирован алгоритм. Этот фактор оказывает исключительно большое влияние на эффективность разрабатываемого алгоритма.

3. **Приближенность представления вещественных чисел** в памяти компьютера. Это требует, чтобы при разработке алгоритма всюду, где производится сравнение вещественных значений, использовался некоторый задаваемый программистом уровень точности. Игнорирование специфики машинной арифметики является распространенной ошибкой при разработке алгоритма. Например, проверка принадлежности трех точек одной прямой, программируется обычно следующим образом: в уравнение прямой, проходящей через две точки, подставляются координаты третьей точки. Все точки лежат на одной прямой тогда и только тогда, когда при подстановке получается нуль. Однако из-за неточности машинной арифметики при выполнении такой проверки на ЭВМ нуль почти наверняка никогда не будет получен, даже если теоретически названное свойство выполнено. Реально приходится считать свойство выполненным, когда полученный при подстановке результат меньше (по модулю) некоторого **предусмотренного** разработчиком числа, например 10^{-8} .

Обоснование алгоритма

Строгое математическое доказательство правильности работы алгоритма – обычно очень трудная задача, главным образом из-за того, что трудно показать правильность работы циклов и рекурсивных процедур. Вместе с тем демонстрация правильной работы алгоритмов на некотором наборе тестов еще не означает, что он всегда будет работать правильно. Следует помнить, что различных комбинаций входных данных бывает, как правило, бесконечно много (или «практически» бесконечно много). Поэтому необходимо сопровождать алгоритм некоторым рассуждением, которое, даже не будучи строгим доказательством, в достаточной мере убеждает нас в его правильности. Конечно, рассуждение в стиле: «алгоритм перебирает все варианты, поэтому он правилен» не выдерживает критики, так как в этом случае возникает очевидный вопрос, а как убедиться, что алгоритм действительно перебирает все варианты.

Наилучшим практическим подходом к обоснованию алгоритма является, пожалуй, его обоснование «*по построению*», когда используется описанный выше метод *пошаговой разработки*. Чтобы получить **правильный** алгоритм, необходимо следить за **правильностью** детализации его шагов в ходе такого построения. Но это уже значительно более простая задача, так как детализация шага, как правило, происходит в соответствии с определением того, что он должен делать.

При таком подходе *построение* алгоритма и его *обоснование* тесно переплетаются друг с другом. При этом следует, конечно, понимать, что если на этапе анализа задачи был выбран неверный подход к ее решению, то даже самая аккуратная последующая детализация исходной спецификации уже не позволит получить правильный алгоритм.

Обоснование алгоритма будет выглядеть еще более убедительно, если оно дополнено доказательствами, позволяющими убедиться в правильности фрагментов программы, например, некоторых циклов. Ниже приведен пример, иллюстрирующий сущность таких доказательств.

Пример. Рассмотрим простую функцию, реализующую возведения целого неотрицательного числа X в целую неотрицательную степень N (рис. 5).

```
int POWER(int X, int N){
if ((X<0) || (N<0)) cerr<<"POWER: (X<0) || (N<0)"<<'\\n';
exit(-1);
if (N==0) if (X!=0) return(1); else {
    cerr<<"Ошибка в POWER"<<'\\n'; exit(-1);
}
else if (X==0) return(0);
int RES=1;
while (N-->0) RES*=X;
return (RES);
};
```

Рис. 5. Функция возведения целого числа X в целую степень

Если в функцию переданы некорректные аргументы, ее выполнение завершается передачей сообщения с помощью выходного потока `cerr`. Некорректными для данной функции будут отрицательные значения N и X . Правильность работы алгоритма, когда один из входных параметров функции равен нулю, следует из определения степени, и обеспечивается программой. Если же все входные параметры являются положительными числами, то результат будет получен в переменной `RES` после выполнения оператора **while**.

Докажем, что при каждом прохождении условия продолжения цикла значения переменных X , N и `RES` связаны соотношениями $X=X_0$, $0 < N < N_0$, $RES = X^{N_0-N}$, где X_0 и N_0 – начальные значения переменных X и N , присвоенные им оператором ввода. Другими словами, мы покажем, что конъюнкция (объединение) этих условий является так называемым инвариантом (неизменным параметром) цикла. Доказательство проведем методом математической индукции. Через X^* , N^* и `RES` будем обозначать значения X , N , `RES` при k -м прохождении цикла **while**.

1. **Базис индукции.** Когда впервые управление попадает в цикл **while**, имеем $X_i = X_0$, $N_i = N_0$ и $RES_i = 1 = X^{N_0-N}$, т.е. требуемые соотношения выполнены.

2. **Шаг индукции.** Пусть утверждение выполнено для первых k прохождений цикла **while**, и управление передается в тело цикла $(k+1)$ -й раз. Это может случиться лишь в том случае, когда $N_k > 0$, иначе, если бы $N_k = 0$, то уже после k -го прохождения цикл был бы завершен, а управление передано на оператор **return** и вновь в тело цикла не попало бы. Но если $N_k > 0$, то после k -го прохождения выполняются операторы, которые стоят в теле цикла. Затем управление вновь, т. е. в $(k+1)$ -й раз, передается циклу **while**. Таким образом, значение X за это время не изменилось, т. е. $X_{k+1} = X_0$, а значения N и `RES` изменились так, что $N_{k+1} = N_k - 1$, т. е. $0 \leq N_{k+1} \leq N_0$ (в силу предположения индукции $N_k > 0$), и $RES_{k+1} = RES_k * X = X^{N_0-N_k} * X$ (в силу предположения индукции) $= X^{N_0-(N_k-1)} = X^{N_0-N_{k+1}}$, что и требовалось доказать.

Теперь ясно, что в тот момент, когда при проверке условия продолжения цикла **while** значение N окажется равным нулю (такой момент наступит, так как N уменьшается в цикле), мы, в силу доказанного, будем иметь $RES = X_0^{N_0}$. Поэтому оператор **return**, которому в этом случае передается управление, возвратит правильный результат.

В заключение отметим, что язык C++ обладает средствами, позволяющими включать утверждения (такие, как инварианты циклов) непосредственно в текст программы и соответственно контролировать по ходу ее исполнения правильность работы тех или иных конструкций.

Средства повышения надежности алгоритма

Описанный выше метод пошаговой разработки ориентирован на создание *надежных* алгоритмов, поскольку он позволяет разработчику все время контролировать *управляющую и информационную* структуру алгоритма. Тем не менее, сам по себе пошаговый метод не может гарантировать правильности алгоритма. Ошибки, особенно при разработке нетривиальных алгоритмов, неизбежно возникают, например, при вводе текста программы. Поэтому с самого начала необходимо применять дополнительные средства (так называемые *стопоры ошибок*), повышающие надежность алгоритма и позволяющие эффективнее вести поиск ошибок при последующей отладке. Одно из таких средств было упомянуто в конце предыдущего раздела – это вставка утверждений в различные точки С++-программ, прежде всего в операторы цикла. Отметим еще несколько полезных средств.

1. Необходимо распечатать значения входных данных сразу после их ввода. Это позволяет контролировать правильность вводимой информации (и не только студенту, но и преподавателю).

2. При входе в процедуру полезно распечатывать ее имя, а также имена и значения всех фактических параметров (за исключением параметров-переменных: их значения стоит распечатывать перед выходом из процедуры).

3. Следует предусмотреть печать сообщений о прохождении так называемых *узловых точек* программы. Точка считается узловой, если управление проходит через нее при любых начальных данных. Фактически узловые точки разбивают алгоритм на части, выполняющие некоторые самостоятельные и законченные функции.

4. Наконец, важным средством повышения надежности С++-программ является использование так называемых *счетчиков*, позволяющих контролировать число прохождений различных ветвей программы.

Выбор и обоснование набора тестов

С разработкой алгоритма и программы тесно связано и построение набора тестов, на которых ее работа будет проверяться во время отладки. При совместной разработке программ и набора тестов легче добиться полноты этого набора, т. е. ситуации, когда каждый имеющийся в программе переход был пройден на одном из тестов набора.

Сразу отметим, что *тест* представляет собой не только некоторую *совокупность входных данных* программы, но и *точное описание всех результатов*, которые программа должна выработать на этих данных, причем в том виде, как эти результаты должны быть расположены на выдаче. Естественно, что в понятие «результат» включаются и те сообщения, которые должны вывести операторы печати, размещенные в программе для повышения ее надежности. Нарушение этого требования приводит к тому, что оценка правильности работы программы на выбранных данных становится невозможной.

Например, если речь идет о программе, проверяющей свойство простоты натурального числа N (см. пример ниже), то нет смысла тестировать ее для $N = 1234567$, если заранее не известно является ли это число простым или составным.

Обычно уже из формулировки задачи вытекает необходимость проверки работы программы хотя бы на двух тестах. Пусть, например, в задаче требуется подсчитать количество окружностей, каждая из которых проходит хотя бы через три различные точки из заданного множества, в котором не меньше трех точек.

Тогда в качестве тестов заведомо необходимо взять:

- множество точек, лежащих на одной прямой (с ожидаемым сообщением об отсутствии искомым окружностей);
- множество, в котором не все точки лежат на одной прямой (в этом случае тест должен содержать ответ – сколько требуемых окружностей должна обнаружить программа с учетом приближенности вычислений).

Если алгоритм, решающий задачу, имеет разветвление, набор тестов необходимо пополнить так, чтобы иметь возможность *пройти каждую из ветвей*.

Аналогично, если встречается оператор цикла с условием продолжения, то в наборе должен появиться *тест, на котором тело цикла не выполняется ни разу*, а также *тест, на котором тело цикла выполняется хотя бы один раз*.

Проиллюстрируем построение набора тестов на простом примере,

Пример. Пусть требуется построить программу, которая печатает сообщение N – ПРОСТОЕ, если натуральное число N является простым, и сообщение N – СОСТАВНОЕ в противном случае.

Сразу же отметим, что в наборе тестов должно быть хотя бы одно простое и одно составное число. Далее, число N по определению простое, если оно делится лишь на себя и на единицу; но при этом единица по определению – число составное. Такое определение разбивает все числа на две группы: 1 (особый случай) и остальные числа. Значит, число 1 заведомо должно быть среди тестов для разрабатываемого алгоритма. Далее, простоту N можно проверить, последовательно разделив его на числа от 2 до $\sqrt{N}$, если хотя бы одно такое деление удалось, то N – составное, иначе N – простое.

С точки зрения эффективности алгоритма предпочтительно разбивать числа на четные и нечетные; при этом четные разбиваются в свою очередь на две группы – число 2 (простое) и остальные (составные). Делители нечетного N можно искать среди нечетных чисел от 3 до $\sqrt{N}$. Отсюда ясно, что в наборе тестов должны присутствовать число 2, некоторое четное число, отличное от 2, и хотя бы два нечетных числа – простое и составное. На рис. 6 приведена программа для решения этой задачи.

В этой программе тело оператора цикла выполняется для всех нечетных $N > 3$, но не выполняется для $N = 3$. Значит, число 3 также нужно включить в набор тестов. Окончательный набор тестов мог бы выглядеть следующим образом:

{(1, «1 – СОСТАВНОЕ»), (2, «2 – ПРОСТОЕ»), (3, «3 – ПРОСТОЕ»), (6, «6 – СОСТАВНОЕ»), (11, «11 – ПРОСТОЕ»), (77, «77 – СОСТАВНОЕ»)}.

Отметим, что в ходе отладки программы в ней выявляются ошибки и вносятся различные изменения; в частности, могут появиться новые разветвления. При этом следует помнить о том, что все изменения требуют надлежащим образом модифицировать и набор тестов. Кроме того, в ходе отладки набор может пополняться тестами, позволяющими локализовать причины обнаруживаемых ошибок.

```
#include <iostream.h> //
#include <math.h>      // математические функции
int main () {
    cout << "Программа определит является введенное число N
    простым или составным" <<'\n'<<"Введите N ";
    cin>>x;
    if (N==1) cout << N << " - составное";
    else if (N==2) cout << N << " - простое";
    else if (!(N%2)) cout << N << " - составное";
    else {
        int sqrtN=sqrt(N);
        for (del=3; del+=2; del< sqrtN){
            if (N%del) continue;
            else {
                cout << N << " - составное";
                break;
            };
        };
    };
    return(0);}
```

Рис. 6. Программа, проверяющая принадлежность натурального числа к простым числам

Описание разработанных алгоритмов

Прежде всего, нужно иметь в виду, что такое описание предназначено не для машины, а для человека. Другими словами, речь идет о некотором тексте (т. е. о словесном описании), по которому можно получить представление об общей структуре разрабатываемого алгоритма, о смысле его отдельных шагов и их логической взаимосвязи. Сохранение достаточно высокого уровня описания алгоритма также облегчает его обоснование. Поэтому шаги алгоритма должны описываться в терминах тех объектов и отношений между ними, о которых идет речь в формулировке задачи.

Например, для «геометрической» задачи шаги алгоритма следует описывать как действия над точками, прямыми и т. п., как проверки свойств типа принадлежности трех точек одной прямой и т. п. В тексте не должно быть фраз, заде-

вающих коды этих объектов, например не должна упоминаться матрица координат точек некоторого множества.

Это, конечно, не исключает использования математической и иной удобной символики. Кроме того, структура алгоритма станет более ясной, если шаги записывать в виде, напоминающем конструкции языков программирования, такие, как, например, операторы цикла и условные операторы (см. ниже пример отчета, а также пример, приведенный выше для иллюстрации метода пошаговой разработки).

Выбор представления данных

Данные, обрабатываемые алгоритмом, делятся на входные, промежуточные и выходные. Специфика входных и выходных данных состоит в том, что с ними имеет дело не только алгоритм, но и пользователь программы. Поэтому различают две формы представления таких данных – внешнее и внутреннее.

Основное требование к внешнему представлению данных состоит в его максимальном удобстве, понятности и естественности для пользователя, чтобы дать возможность пользователю достаточно легко подготовить данные для ввода и оценить результат по выходным данным.

Например, если в качестве входных данных выступает некоторое множество точек на плоскости, то внешнее представление может выглядеть как последовательность чисел – координат этих точек; при этом естественно, чтобы сначала шли координаты первой точки, затем второй и т. д. И наоборот, неестественно требовать, чтобы в этой последовательности сначала шли все абсциссы, а затем уже все ординаты, или чтобы сначала обязательно шли координаты точек из верхней полуплоскости и только затем из нижней. Это потребовало бы от пользователя непростой предварительной работы *по упорядочению* исходной информации в соответствии с этими требованиями.

Точно так же недопустим такой вывод полученной информации, который требовал бы от пользователя нетривиальных усилий по переводу напечатанных данных в понятия исходной постановки задачи. Необходимо также помнить о содержательном группировании распечатываемых данных на бумаге – так, чтобы, скажем, при выводе матрицы ее элементы располагались по строкам, а не в виде сплошной последовательности.

Далее необходимо зафиксировать *внутреннее* представление входных, выходных и промежуточных данных, обрабатываемых в программе, т.е. выбрать те средства языка программирования, которыми описывается это представление. Отметим, что для входных данных оно *вовсе не обязано совпадать с внешним представлением*.

Например, в качестве внутреннего представления множества из N точек на плоскости удобно брать либо матрицу размером $N \times 2$, в которой n -я строка состоит из координат n -й точки, либо в виде массива, элементами которого являются структуры с двумя полями – абсциссой и ординатой соответствующей точки. Вообще выбор внутреннего представления определяется главным образом

требованиями эффективности того алгоритма, который нужно построить для решения задачи. Это может оказать существенное влияние, как на объем используемой памяти, так и на время работы алгоритма.

Ниже перечислим конкретные рекомендации по выбору внешнего и внутреннего представлений входных данных, используемых в формулировках заданий.

Для описания внешнего представления исходных данных, соответствующих применяемым в формулировках заданий понятиям, рекомендуется использовать синтаксические формулы. Для описания внутреннего представления этих понятий применяются конструкции языка C++. Константы, встречающиеся в этих описаниях, следует рассматривать как ограничения на размер входных данных, представляющих соответствующие понятия.

1. Множество точек на плоскости

Внешнее представление:

$$\begin{aligned}\langle \text{множество точек} \rangle &\triangleq \langle \text{количество точек} \rangle \langle \text{пробелы} \rangle \{ \text{точка} \}^* \\ \{ \text{точка} \}^* &\triangleq \langle \text{точка} \rangle | \langle \text{точка} \rangle \{ \text{точка} \}^* \\ \langle \text{количество точек} \rangle &\triangleq \langle \text{целое число} \rangle, \\ \langle \text{точка} \rangle &\triangleq \langle \text{пробелы} \rangle \langle \text{абсцисса} \rangle \langle \text{пробелы} \rangle \langle \text{ордината} \rangle, \\ \langle \text{абсцисса} \rangle &= \langle \text{ордината} \rangle \triangleq \langle \text{вещественное число} \rangle.\end{aligned}$$

Внутреннее представление:

```
const int PointNumber=200;    // Количество точек
struct point {                // Структура point
    double X, // Абсцисса
    double Y  // Ордината
};
point PointSet[PointNumber]; // Массив структур point
```

Перевод из внешнего представления во внутреннее может выглядеть, например, так:

```
void Inp (char*FName){
    ifstream f(FName); if(!f) exit(-1);
    int PN;
    f>> PN; cout<<"Задано "<<PN<<" точек"<<'\\n';
    if (PN>PointNumber) exit(-1);
    for(int i=0; i<PN; i++){
        f>> PointSet[i].X >> PointSet[i].Y;
        cout<<"X="<<PointSet[i].X
            <<" Y="<<PointSet[i].Y<<'\\n';
    }
}
```

Отметим, что вместе с таким переводом мы одновременно печатаем вводимые данные, т. е. делаем и обратный перевод.

Аналогично строятся внешнее и внутреннее представления для множества точек в трехмерном пространстве, а также и для множества материальных точек.

2. Прямая на плоскости

Внешнее представление.

Вариант 1:

$$\langle \text{прямая1} \rangle \triangleq \langle X \text{ коэффициент} \rangle \langle \text{пробелы} \rangle \langle Y \text{ коэффициент} \rangle \langle \text{пробелы} \rangle \langle \text{Свободный член} \rangle, \\ \langle X \text{ коэффициент} \rangle = \langle Y \text{ коэффициент} \rangle = \langle \text{Свободный член} \rangle \triangleq \langle \text{вещественное число} \rangle.$$

Здесь заданы коэффициенты линейного уравнения прямой

$$A \cdot X + B \cdot Y + C = 0.$$

Разумеется, при этом должно выполняться условие $|A| + |B| \neq 0$.

Вариант 2:

$$\langle \text{прямая2} \rangle \triangleq \langle \text{точка} \rangle \langle \text{пробелы} \rangle \langle \text{точка} \rangle \langle \text{пробелы} \rangle.$$

В этом варианте прямая представляется парой (различных!) точек, через которые она проходит.

Внутреннее представление:

```
struct line1 {
    double A,          // X коэффициент
    double B,          // Y коэффициент
    double C           // X коэффициент
};

struct line2 {
    point First,       // Первая точка
    point Second      // Вторая точка
};
```

3. Окружность

Внешнее представление:

$$\langle \text{окружность} \rangle \triangleq \langle \text{центр} \rangle \langle \text{пробелы} \rangle \langle \text{радиус} \rangle \langle \text{пробелы} \rangle, \\ \langle \text{радиус} \rangle \triangleq \langle \text{вещественное число} \rangle, \\ \langle \text{центр} \rangle \triangleq \langle \text{точка} \rangle.$$

Здесь нужно требовать положительности радиуса.

Внутреннее представление:

```
struct circle {
    point centre,      // центр
    double r,          // радиус
};
```

Например, если множество точек на плоскости представлено массивом типа `PointSet`, то каждую точку множества можно однозначно идентифицировать, указывая ее порядковый номер в массиве. Тогда внутренним представлением треугольника или четырехугольника могут служить значения трех или, соответственно, четырех переменных – номеров вершин многоугольника.

3. Последовательность

Внешнее представление последовательности строится с использованием следующего общего синтаксического правила:

$$\langle \text{последовательность} \rangle \triangleq \{ \langle \text{элемент} \rangle \langle \text{разделитель} \rangle \}.$$

Конкретные внешние представления получаются после фиксации правил для представлений элемента и разделителя последовательности. Например, используя правило

$$\langle \text{разделитель} \rangle \triangleq \langle \text{пусто} \rangle,$$

получаем способ внешнего представления последовательности без разделителя. Для последовательности целых чисел естественно взять

$$\begin{aligned} \langle \text{разделитель} \rangle &\triangleq \langle \text{пробелы} \rangle, \\ \langle \text{элемент} \rangle &\triangleq \langle \text{целое число} \rangle. \end{aligned}$$

Внутреннее представление последовательности может совпадать с внешним, если в алгоритме обработки ее элементов не требуется многократно обращаться к обработке одного и того же элемента.

В противном случае в качестве внутреннего представления можно использовать, например, запись

```
const int MaxLen =500;
ELEMENTYPE Sequence[MaxLen]; // последовательность
                               //элементов типа ELEMENTYPE
```

Ясно, что в этом случае `ELEMENTYPE` должно быть предварительно определено с помощью операторов **struct**, **class** или **typedef**.

4. Текст

Внешнее представление:

$$\langle \text{текст} \rangle \triangleq \langle \text{символ} \rangle \{ \text{символ} \}.$$

Внутреннее представление. Когда не требуется многократно возвращаться к рассмотрению одного и того же вхождения символа, внутреннее представление может совпадать с внешним, в противном случае в качестве внутреннего представления можно взять, например

```
const int MaxLen =1500;
char text[MaxLen]; // последовательность символов
```

Слова и предложения можно представлять как тексты. Иногда может оказаться целесообразным внутреннее представление предложения как двумерного массива.

```
const int MAXWORDLEN = 20;
const int MAXWORDNUMB = 50;
typedef char WORD[MAXWORDLEN];
WORD SENTENCE [MAXWORDNUMB];
```

Кодирование алгоритма (программирование)

Записать разработанный алгоритм на языке программирования можно различными способами. При этом получаются программы, которые отличаются:

- по степени наглядности и удобству внесения изменений,
- по длине и объему используемой памяти,
- по быстродействию и многим другим характеристикам.

Эти характеристики определяют качество программы.

Перечислим некоторые рекомендации, направленные на улучшение качества разрабатываемых программ.

Структурное программирование и наглядность программ. Прежде всего, нужно иметь в виду, что написание программы представляет собой фактически продолжение разработки алгоритма с дальнейшей детализацией его шагов и фиксацией на языке программирования. В результате этого шаги становятся элементарными.

Поэтому здесь следует применять тот же метод пошаговой разработки, описанный выше. Этот метод ориентирован на получение, в конечном счете, **структурированных** программ, *структура* текста которых отражает *структуру* вычислений.

Для того чтобы получить структурированную программу, рекомендуется на всех этапах уточнения использовать управляющие структуры языка, имеющие один вход и один выход (условные операторы, операторы выбора, операторы цикла). В то же время следует ограничить употребление оператора перехода. Тогда получаемая программа наглядна, легко читается и понимается не только автором, но и другими программистами (в частности, преподавателем).

В качестве иллюстрации рассмотрим два фрагмента программ для поиска индексов первого ненулевого элемента в одномерном массиве:

```
for (i=0; i<100; i++) if (A[i]) break;
```

и

```
i = 0;
2: if ( A[i]!=0) goto 1;
   else{
       i = i + 1;
       if i<= 100 goto 2;
   };
1:
```

С точки зрения структурированности первый из фрагментов следует предпочесть второму.

Здесь уместно отметить, что на возможность создания структурированных программ существенное влияние оказывает имеющийся в языке набор средств управления:

- *Pascal* хорошо ориентирован на структурное программирование;
- *Fortran*, вынуждает программиста использовать большое количество операторов перехода;
- *C++* также хорошо ориентирован на структурное программирование, т.к. дает возможность явно выделить циклическую структуру или указать альтернативы при разветвлении.

Дадим еще несколько рекомендаций по улучшению наглядности программного текста.

Рекомендуется с помощью **сдвига начала строк** выделять содержательно разные части и части, относящиеся к синтаксически разным единицам программного текста. Это дает возможность дополнить структурированность управления в программе структурированностью самого ее текста.

Удобно показать вложенность одних конструкций в другие. При этом следует записывать символ «}» под соответствующим «{», а «else» под соответствующим оператором «if» и т.д. Этот прием позволяет облегчить не только чтение программы, но и поиск в ней некоторых ошибок, например места нарушения баланса скобок «{...}».

Использование комментариев и содержательно осмысленных (мнемонических) имен – также весьма простой и мощный прием в деле улучшения наглядности программы.

В связи с этим укажем на весьма распространенное для начинающих программистов явление – **употребление только одно- или двухбуквенных имен** в своей программе. В результате такие интуитивно родственные имена, как, например А, В, С, даются не объектам, имеющим родственный смысл, а, скажем, массиву, индексу элемента в этом массиве и процедуре. Это крайне затрудняет чтение и, следовательно, понимание программы.

Что касается **комментариев**, то они должны кратко и точно пояснять смысл тех шагов алгоритма, к которым они относятся. Не следует делать их слишком частыми или слишком длинными.

Нужно также разумно **ограничить длину строк** в программе. Дело в том, что программа должна допускать простую и эффективную модификацию, т.е. изменение в целях исправления ошибок, для нового использования программы или улучшения ее качества.

Рекомендации по улучшению качества программ

Перечислим только самые важные приемы, с помощью которых можно добиться улучшения качества программ.

1. Необходимо *описывать многократно используемые фрагменты программы в виде процедур*. Это весьма мощный прием в программировании. Суть его состоит в том, что некоторый участок программы оформляется в виде функции. Таким образом, если результаты работы такого участка неоднократно используются в программе (возможно, с разными значениями параметров), то за счет замены этих участков вызовами соответствующей функции достигается сокращение длины программы, повышаются ее структурированность и надежность, облегчается процесс написания и отладки.

2. Если значение некоторого выражения используется в программе несколько раз, причем ни одна из переменных при этом не изменяет своего значения, целесообразно занести значение этого выражения во *вспомогательную переменную* и затем использовать ее. Например, вместо

$$X := (2 * Y + A * B) * (A * B * C + 2 * Y) ;$$

лучше написать

$$Y1 := 2 * Y ; \quad A1 := A * B ; \quad X := (Y1 + A1) * (A1 * C + Y1) ;$$

3. *Выражения, значения которых не изменяются* при выполнении тела цикла, следует вычислять заранее, и перед оператором цикла присваивать вспомогательной переменной соответствующее значение.

Следует иметь в виду, что два последних приема повышают эффективность программы, но могут понизить ее наглядность.

Варианты задач для начинающих

Ниже приведены тексты заданий для начинающих, которые могут быть предложены на начальном этапе изучения программирования. Для решения этих задач студенту достаточно знания основных конструкций языка C++.

Общие требования

Программа считается принятой преподавателем, если проходят все тесты, у преподавателя нет серьезных замечаний по алгоритму и организации ввода-вывода, и студент ответит на все вопросы по тексту программы. Число попыток сдачи программы не ограничено. Возможна помощь преподавателя на этапе разработки программы в виде консультации, ответов на возникшие вопросы, но не разработки алгоритма вместо студента.

Программы необходимо снабжать комментариями: функциональное назначение программы (условие задачи), описание входных данных и результата.

Пример плохого стиля оформления полей ввода/вывода данных

```
Введите N: 23
Введите K: 9
7
Нажмите любую клавишу для продолжения
```

Пример оформления полей ввода/вывода данных

```
Программа вычисляет время, необходимое для преодоления
расстояния S со скоростью V.
Введите расстояние (км): 120
Введите скорость передвижения (км/ч): 60
Требуется 2 часа (ов).
Нажмите любую клавишу для продолжения.
```

Вариант 1

Линейные алгоритмы

1. Угол A задан в градусах, минутах и секундах. Найти его величину в радианах (с максимально возможной точностью). Программа должна быть работоспособной для углов больше развернутых, а также для отрицательных углов.

Разветвляющиеся алгоритмы

2. На шахматной доске стоят три ферзя (ферзь бьет по вертикали, горизонтали и диагонали), позиции которых заданы в виде колонка+строка, например, F6. Найти те пары из них, которые угрожают друг другу.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + b & \text{при } x < 0 \text{ и } b \neq 0, \\ \frac{x-a}{x-c} & \text{при } x > 0 \text{ и } b = 0, \\ \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Численно убедиться, является ли заданная функция $y = f(x)$ четной или нечетной на заданном отрезке $-a \leq x \leq a$. Учесть погрешность вычислений и возможные точки разрыва функции. Проверить, например, для функций $y = x^4$, $y = \cos x$, $y = e^x$, вычисляя их на некотором отрезке с заданным шагом. Шаг вычисления и границы отрезка вводятся с клавиатуры, функция, которая проверяется на четность – должна быть введена в программе.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- сумму отрицательных элементов массива;
- произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- количество строк, не содержащих ни одного нулевого элемента;
- максимальное из чисел, встречающихся в заданной матрице более одного раза.

Прикладные математические задачи

7. Натуральное число в p -ичной системе счисления задано своими цифрами, хранящимися в массиве $K(n)$. Проверить корректность такого представления и перевести число в q -ичную систему (возможно, число слишком велико, чтобы получить его внутреннее представление, кроме того $p \leq 10, q \leq 10$).

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 2

Линейные алгоритмы

1. Угол A задан в радианах. Найти его величину в градусах, минутах и секундах. Программа должна быть работоспособной для углов больше развернутых, а также для отрицательных углов.

Разветвляющиеся алгоритмы

2. На шахматной доске стоят черный король и белые ладья и слон (ладья бьет по горизонтали и вертикали, слон – по диагоналям). Проверить, есть ли угроза королю и если есть, то от кого именно. Учесть возможность защиты (например, ладья не бьет через слона).

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} \frac{1}{ax} - b & \text{при } x + 5 < 0 \text{ и } c = 0, \\ \frac{x - a}{x} & \text{при } x + 5 > 0 \text{ и } c \neq 0, \\ \frac{10x}{c - 4} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Утверждается, что функция $y = f(x)$ – периодическая с периодом T . Проверить это численно, вычислив функцию с постоянным шагом на отрезке $[0; 5T]$. Учесть погрешность вычислений и возможные точки разрыва функций. Проверить на примере функций $y = \sin^2 x$, $y = \operatorname{tg} x$, $T = \pi$, $y = \frac{\sin x}{x}$, $T = 2\pi$. Шаг вы-

числения вводится с клавиатуры, функция, которая проверяется на периодичность – должна быть введена в программе.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- сумму положительных элементов массива;
- произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по возрастанию.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

Прикладные математические задачи

7. Для натуральных чисел, не превосходящих заданного k , проверить признак делимости на 9 (сумма цифр числа, делящегося на 9, также делится на 9). Распечатать m последних таких чисел ($m \ll k$).

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 3

Линейные алгоритмы

1. Длина отрезка задана в дюймах (1 дюйм=2,54см). Перевести значение длины в метрическую систему, то есть выразить ее в метрах, сантиметрах и миллиметрах. Например, 21 дюйм = 0 м 53 см 3,4 мм.

Разветвляющиеся алгоритмы

2. На шахматной доске стоят черный король и три белые ладьи (ладья бьет по горизонтали и по вертикали). Проверить, не находится ли король под боем, а если есть угроза, то от кого именно.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + bx + c & \text{при } a < 0 \text{ и } c \neq 0, \\ \frac{-a}{x - c} & \text{при } a > 0 \text{ и } c = 0, \\ a(x + c) & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$ вводятся с клавиатуры.

4. Предприниматель, начав дело, взял кредит размером k рублей под p процентов годовых и вложил его в свое дело. По прогнозам, его дело должно давать прибыль r рублей в год. Сможет ли он накопить сумму, достаточную для погашения кредита, и если да, то через сколько лет. Все параметры вводятся с клавиатуры.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- произведение элементов массива с четными номерами;
- сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные нулю, считать положительными).

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- количество столбцов, содержащих хотя бы один нулевой элемент;
- номер строки, в которой находится самая длинная серия одинаковых элементов.

элементов.

Прикладные математические задачи

7. Число делится на 11, если разность между суммой цифр, стоящих на нечетных местах, и суммой цифр, стоящих на четных местах, кратна 11. Проверить этот признак для всех натуральных чисел, не превосходящих заданного m , и вывести числа, кратные 11.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 4

Линейные алгоритмы

1. Заданы моменты начала и конца некоторого промежутка времени в часах, минутах и секундах (в пределах одних суток). Найти продолжительность этого промежутка в тех же единицах измерения.

Разветвляющиеся алгоритмы

2. В некотором университете принято, что старшая цифра номера студенческой группы означает номер факультета, средняя – последнюю цифру года поступления, младшая – порядковый номер группы на курсе. Продолжительность обучения – не более 6 лет. Дан номер группы студента этого университета. Напечатать, в каком году он поступил и на каком факультете учится. Например, группа 432, факультет математический, год поступления 1993. Для справки приведены номера факультетов: 1 – исторический; 2 – экономический; 3 – юридический; 4 – математический; 5 – физический; 6 – химический; 7 – биологический; 8 – филологический; 9 – географический.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} -ax - c & \text{при } c < 0 \text{ и } x \neq 0, \\ \frac{x - a}{-c} & \text{при } c > 0 \text{ и } x = 0, \\ \frac{bx}{c - a} & \text{в остальных случаях.} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Каждая из деталей должна последовательно пройти обработку на каждом из трех станков. Продолжительности обработки каждой детали на каждом станке вводятся группами по 3 числа, до исчерпания ввода в порядке поступления деталей на станок. Сколько времени займет обработка всех деталей.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- сумму элементов массива с нечетными номерами;
- сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- произведение элементов в тех строках, которые не содержат отрицательных элементов;
- максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

Прикладные математические задачи

7. В массиве $A(m)$ хранятся различные вещественные числа (как больше, так и меньше единицы). Округлить их, оставив в каждом по 3 значащих цифры.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 5

Линейные алгоритмы

1. 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? (Значения k , p и стоимость магазина вводить с клавиатуры).

Разветвляющиеся алгоритмы

2. Суточный рацион коровы составляет u кг зерна, v кг силоса и w кг комбикорма. В хозяйстве, содержащем стадо из k голов, осталось s центнеров сена, t тонн силоса и f кг мешков комбикорма по 50 кг. Сколько еще дней хозяйство сможет кормить коров по полному рациону? Какой из кормов кончится раньше других?

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} a - \frac{x}{10 + b} & \text{при } x < 0 \text{ и } b \neq 0, \\ \frac{x - a}{x - c} & \text{при } x > 0 \text{ и } b = 0, \\ 3x + \frac{2}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Материальная точка бросается на горизонтальную плоскость по углом α к ней со скоростью v_0 . При каждом ударе о плоскость кинетическая энергия точки уменьшается в β раз. Найти абсциссы первых n точек касания. Сопротивлением воздуха пренебречь.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- сумму отрицательных элементов массива;
- произведение элементов массива, расположенных между максимальным и минимальным элементами.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- сумму элементов в тех столбцах, которые не содержат отрицательных элементов;
- максимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

Прикладные математические задачи

7. Для заданного m получить таблицу первых m простых чисел.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 6

Линейные алгоритмы

1. 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? (Значения, k, p, s, n вводятся с клавиатуры).

Разветвляющиеся алгоритмы

2. Можно ехать на такси со скоростью v км/час и оплатой p руб./км, либо идти пешком со скоростью w км/ч бесплатно. Как с наименьшими затратами преодолеть путь s за время t , если это возможно? Каковы эти затраты? Когда времени слишком мало, чтобы успеть даже на такси, вывести текст: «Не успеть!»

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + b^2x & \text{при } c < 0 \text{ и } b \neq 0, \\ \frac{x+a}{x+c} & \text{при } c > 0 \text{ и } b = 0, \\ \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Суточный рацион коровы составляет u кг сена, v кг силоса и w кг комбикорма. В хозяйстве, содержащем стадо из k голов, осталось s кг сена, t кг силоса и f кг комбикорма. В стаде ежедневно погибает $p\%$ коров; ежедневно $q\%$ оставшегося сена сгнивает; $r\%$ силоса разворовывается колхозниками; $t\%$ комбикорма распродает заведующий фермой. Когда нельзя будет кормить всех оставшихся коров по полному рациону? Какой из видов кормов кончится раньше других?

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- минимальный элемент массива;
- сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом все остальные.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент;
- номера строк и столбцов всех седловых точек матрицы.

Матрица A имеет седловую точку A_{ij} , если A_{ij} является минимальным элементом в i -й строке и максимальным в j -ом столбце.

Прикладные математические задачи

7. Своими цифрами в массивах $K(m)$ и $L(n)$ заданы два целых числа в p -ичной системе счисления ($p \leq 10$). Найти в таком же виде их сумму, не вычисляя самих чисел.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 7

Линейные алгоритмы

1. Заданы три корня кубического уравнения a, b, c . Найти коэффициенты этого уравнения.

Разветвляющиеся алгоритмы

2. Как известно, число делится на 3 тогда и только тогда, когда сумма его цифр делится на 3. Проверить этот признак на примере вводимого трехзначного числа.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} -ax^2 - b & \text{при } x < 5 \text{ и } c \neq 0, \\ \frac{x-a}{x} & \text{при } x > 5 \text{ и } c = 0, \\ \frac{-x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Известно время начала и окончания (например с 6:00 до 24:00) работы некоторого пригородного автобусного маршрута с одним автобусом на линии, а также протяженность маршрута в минутах (в один конец) и время отдыха на конечных остановках. Составить суточное расписание этого маршрута (моменты отправления с конечных пунктов) без учета времени на обед и пересменку.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- номер максимального элемента массива;
- произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях.

Двумерные массивы

6. Для заданной матрицы размером 8 на 8 найти такие k , что k -я строка матрицы совпадает с k -м столбцом.

Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

Прикладные математические задачи

7. Среди заданных натуральных чисел найти такие, десятичная запись которых не содержит одинаковых цифр.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 8

Линейные алгоритмы

1. Заданы действительная и мнимая части комплексного числа $z=x+iy$. Преобразовать его в тригонометрическую форму и напечатать в виде выражения $z = r(\cos \varphi + i \sin \varphi)$. Для справки: $r = \sqrt{x^2 + y^2}$; $\varphi = \arctg \frac{y}{x}$.

Разветвляющиеся алгоритмы

2. Среди заданных чисел k, l, m найти пары кратных.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} -ax^2 & \text{при } c < 0 \text{ и } a \neq 0, \\ \frac{a-x}{xc} & \text{при } c > 0 \text{ и } a = 0, \\ \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения a , b , c , $X_{\text{нач}}$, $X_{\text{кон}}$, dX вводятся с клавиатуры.

4. Получить таблицу пересчета миль в километры и обратно (1 миля = 1,609344 км) для расстояний, не превышающих k км, с заданным шагом в виде таблицы:

мили		километры
0,6214		1,000.

Простейшие операции с массивами

5. В одномерном массиве, состоящем из N вещественных элементов, вычислить:

- номер минимального элемента массива;
- сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом все остальные.

Двумерные массивы

6. Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик.

Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

Прикладные математические задачи

7. Найти все натуральные числа, не превосходящие заданного m , сумма цифр в десятичном представлении каждого из которых равна заданному k .

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 9

Линейные алгоритмы

1. Заданы уравнения двух пересекающихся прямых на плоскости: $y = k_1x + b_1$, $y = k_2x + b_2$. Найти (в градусах и минутах) угол между ними, используя формулу: $\operatorname{tg} \varphi = \frac{k_2 - k_1}{1 + k_1 k_2}$.

Разветвляющиеся алгоритмы

2. Два отрезка на плоскости заданы координатами своих концов. Определить, имеют ли эти отрезки общие точки.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + b^2x & \text{при } a < 0 \text{ и } x \neq 0, \\ x - \frac{a}{x - c} & \text{при } a > 0 \text{ и } x = 0, \\ 1 + \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$ вводятся с клавиатуры.

4. Для заданных m и x вычислить бином Ньютона $(1 + x)^m$ по формуле разложения в ряд: $(1 + x)^m = \sum_{i=0}^m C_m^i x^i$. Для вычисления C_m^i можно использовать рекуррентное соотношение $C_m^{i+1} = C_m^i \frac{m-i}{i+1}$; $C_m^0 = 1$.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- максимальный по модулю элемент массива;
- сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Двумерные массивы

6. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных ми-

нимумов заданной матрицы размером 10 на 10. Найти сумму модулей элементов, расположенных выше главной диагонали.

Прикладные математические задачи

7. Пусть m натуральных чисел заданы своими цифрами в q -ичной системе счисления, хранящимися в строках матрицы $K(m,n)$. Найти сумму этих чисел в той же системе, не вычисляя самих чисел ($q \leq 10$).

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 10

Линейные алгоритмы

1. Русские неметрические единицы длины: 1 верста=500 сажень; 1 сажень=3 аршина; 1 аршин = 16 вершков; 1 вершок = 44,45 мм. Длина некоторого отрезка составляет p метров. Перевести ее в русскую неметрическую систему.

Разветвляющиеся алгоритмы

2. Лежит ли точка M внутри треугольника, заданного координатами своих вершин A, B, C на плоскости?

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 - bx + c & \text{при } x < 3 \text{ и } b \neq 0, \\ \frac{x-a}{x-c} & \text{при } x > 3 \text{ и } b = 0, \\ \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Проверить численно первый замечательный предел $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$, задавая x значения $1; \frac{1}{2}; \frac{1}{4}; \dots$ до тех пор, пока левая часть равенства не будет отличаться от правой менее чем на заданную погрешность ε (вводится с клавиатуры).

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- минимальный по модулю элемент массива;
- сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях.

Двумерные массивы

6. Соседями элемента A_{ij} в матрице назовем элементы A_{kl} с $i-1 \leq k \leq i+1$; $j-1 \leq l \leq j+1$, $(k,l) \neq (i,j)$. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 10 на 10.

В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали.

Прикладные математические задачи

7. Осуществить циклический сдвиг n -разрядного двоичного представления заданного числа k на m позиций вправо, не находя цифр самого двоичного представления и не пользуясь стандартной процедурой сдвига.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 11

Линейные алгоритмы

1. В равнобедренном прямоугольном треугольнике известна высота h , опущенная на гипотенузу. Найти стороны треугольника.

Разветвляющиеся алгоритмы

2. Можно ли на прямоугольном участке застройки размером a на b метров разместить два дома размерами в плане p на q и r на s метров? Дома можно располагать только параллельно сторонам участка.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + \frac{b}{c} & \text{при } x < 1 \text{ и } c \neq 0, \\ \frac{x-a}{(x-c)^2} & \text{при } x > 1.5 \text{ и } c = 0, \\ \frac{x^2}{c^2} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Проверить численно второй замечательный предел $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$, задавая n значения 1, 2, 3, ... При каком n исследуемое выражение отличается от e меньше, чем на заданную погрешность ε . Погрешность вводить с клавиатуры.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- номер минимального по модулю элемента массива;
- сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Сжать массив, удалив из него все элементы, величина которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

Двумерные массивы

6. Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду.

Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

Прикладные математические задачи

7. Напечатать таблицу сложения одноразрядных чисел в p -ичной системе счисления, $p \leq 16$.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 12

Линейные алгоритмы

1. Найти координаты вершины параболы $y = ax^2 + bx + c$.

Разветвляющиеся алгоритмы

2. Можно ли коробку размером $a \times b \times c$ упаковать в посылку размером $r \times s \times t$? «Углом» укладывать нельзя.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^3 + b^2 + c & \text{при } x < 0.6 \text{ и } b + c \neq 0, \\ \frac{x-a}{x-c} & \text{при } x > 0.6 \text{ и } b + c = 0, \\ \frac{x}{c} + \frac{x}{a} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. В учебном заведении задается начало учебного дня, продолжительность пары или урока, продолжительность обычного и большого перерывов (и их место в расписании), количество пар (уроков). Получить расписание на весь учебный день.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:
 - номер максимального по модулю элемента массива;
 - сумму элементов массива, расположенных после первого положительно-го элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a, b]$, а потом – все остальные.

Двумерные массивы

6. Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

Прикладные математические задачи

7. В заданном вещественном массиве $A(n)$ найти все числа, у которых старшая значащая десятичная цифра есть 9 (числа сильно различаются по величине).

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 13

Линейные алгоритмы

1. Функция $y = \sin x$ на отрезке $[0; 2\pi]$ хорошо аппроксимируется разложением: $y = x - \frac{x^3}{6} + \frac{x^5}{120}$. Для заданного значения аргумента x вычислить y по этой формуле и сравнить с точным значением, вычисленным с помощью стандартной функции `sin`.

Разветвляющиеся алгоритмы

2. Пройдет ли кирпич размером $a \times b \times c$ сквозь прямоугольное отверстие со сторонами r и s ? Стороны отверстия должны быть параллельны граням кирпича.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 + b & \text{при } x - 1 < 0 \text{ и } b - x \neq 0, \\ \frac{x - a}{x} & \text{при } x - 1 > 0 \text{ и } b + x = 0, \\ \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$ вводятся с клавиатуры.

4. Фирма ежегодно на протяжении n лет закупала оборудование стоимостью соответственно $s_1, s_2, \dots, s_n$ рублей в год (эти числа вводятся и обрабатываются последовательно, они не хранятся в массиве, для их ввода использовать одну переменную!) Ежегодно в результате износа и морального старения (амортизации) все имеющиеся оборудование уценяется на $p\%$. Какова общая стоимость накопленного оборудования за n лет?

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество элементов, лежащих в диапазоне от A до B ;

- сумму элементов массива, расположенных после максимального.

Упорядочить элементы массива по убыванию модулей элементов.

Двумерные массивы

6. Осуществить циклический сдвиг элементов прямоугольной матрицы N элементов вправо или вниз (в зависимости от выбора, вводимого с клавиатуры). N может быть больше количества элементов в строке или в столбце.

Прикладные математические задачи

7. Найти все натуральные числа, не превосходящие заданного m и содержащие хотя бы одну девятку в десятичном представлении.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 14

Линейные алгоритмы

1. Найти внутренние углы треугольника ABC , заданного координатами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$ своих вершин на плоскости.

Разветвляющиеся алгоритмы

2. Числа a , b , c тогда и только тогда являются сторонами треугольника, когда существуют такие положительные x , y , z , что

$$\begin{cases} a = x + y; \\ b = y + z; \\ c = x + z. \end{cases}$$

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} -ax^3 - b & \text{при } x + c < 0 \text{ и } a \neq 0, \\ \frac{x - a}{x - c} & \text{при } x + c > 0 \text{ и } a = 0, \\ \frac{x}{c} + \frac{c}{x} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Леспромхоз ведет заготовку деловой древесины. Первоначальный объем ее на территории леспромхоза составлял p кубометров. Ежегодный прирост составляет $k\%$. Годовой план заготовки – t кубометров. Через сколько лет в бывшем лесу будут расти одни опята?

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество элементов массива, равных 0;
- сумму элементов массива, расположенных после минимального.

Упорядочить элементы массива по возрастанию модулей элементов.

Двумерные массивы

6. Осуществить циклический сдвиг элементов квадратной матрицы размерности $M \times N$ вправо на k элементов таким образом: элементы первой строки сдвигаются в последний столбец сверху вниз, из него – в последнюю строку справа налево, из нее – в первый столбец снизу вверх, из него – в первую строку; для остальных элементов – аналогично.

Прикладные математические задачи

7. Десятичное представление заданного натурального числа напечатать вразрядку, т.е. вставить пробелы между цифрами. Число вводить как переменную типа **int**.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 15

Линейные алгоритмы

1. Найти угол (в градусах) между векторами A и B , заданными в трехмерном пространстве, используя формулу:

$$\cos \varphi = \frac{(A, B)}{|A| \cdot |B|} = \frac{x_a x_b + y_a y_b + z_a z_b}{\sqrt{x_a^2 + y_a^2 + z_a^2} \cdot \sqrt{x_b^2 + y_b^2 + z_b^2}}.$$

Разветвляющиеся алгоритмы

2. Треугольник задан координатами своих вершин A, B, C на плоскости. Определить, является он прямо-, остро-, или тупоугольным. Не следует отбрасывать

экстремальные случаи, когда вершины треугольника совпадают или лежат на одной прямой. Например, треугольник с нулевой стороной обладает свойством прямоугольного и имеет два прямых угла!

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} -ax^2 + b & \text{при } x < 0 \text{ и } b \neq 0, \\ \frac{x}{x-c} + 5.5 & \text{при } x > 0 \text{ и } b = 0, \\ \frac{x}{-c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. У гусей и кроликов $2n$ лап. Сколько может быть гусей и кроликов (вывести все возможные сочетания)?

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество элементов массива, больших C ;
- произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом – все положительные (элементы, равные нулю, считать положительными).

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.

Характеристикой строки целочисленной матрицы назовем сумму ее отрицательных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с убыванием характеристик.

Прикладные математические задачи

7. Найти 20 первых троек пифагоровых чисел, т.е. целых k, l, m таких, что $k^2 + l^2 = m^2$.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 16

Линейные алгоритмы

1. 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? (Значения цены литра молока, p и n ввести с клавиатуры).

Разветвляющиеся алгоритмы

2. Треугольник задан длинами своих сторон. Определить, является ли он тупоугольным, прямоугольным или остроугольным.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} a(x+c)^2 & \text{при } x=0 \text{ и } b \neq 0, \\ \frac{x-a}{-c} & \text{при } x=0 \text{ и } b=0, \\ a + \frac{x}{c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Для заданных значений n и x вычислить выражение

$$s = \sin x + \sin \sin x + \dots + \underbrace{\sin \sin \dots \sin x}_{n \text{ раз}}.$$

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество отрицательных элементов массива;
- сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

Двумерные массивы

6. Упорядочить строки целочисленной прямоугольной матрицы по возрастанию количества одинаковых элементов в каждой строке.

Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

Прикладные математические задачи

7. Найти все натуральные числа, не превосходящие заданного n и делящиеся на каждую из своих цифр (в десятичной системе счисления).

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 17

Линейные алгоритмы

1. У квадрата $ABCD$ на плоскости известны координаты двух точек противоположных вершин A и C . Найти координаты точек B и D . Расположение квадрата произвольно, его стороны не обязательно параллельны координатным осям.

Разветвляющиеся алгоритмы

2. Заданы три числа a, b, c . Определить, могут ли они быть сторонами треугольника, и если да, то определить его тип: равносторонний, равнобедренный, разносторонний.

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^2 - cx + b & \text{при } x + 10 < 0 \text{ и } b \neq 0, \\ \frac{x - a}{x - c} & \text{при } x + 10 > 0 \text{ и } b = 0, \\ \frac{-x}{a - c} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Для заданного ε найти наименьшее n такое, что $\frac{2^n}{n!} < \varepsilon$. Вывести все члены последовательности от первого до n -го.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество положительных элементов массива;
- сумму элементов массива, расположенных после последнего элемента, равного нулю.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом все остальные.

Двумерные массивы

6. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2,2), следующий по величине – в позиции (3,3) и т.д., заполнив таким образом всю главную диагональ.

Найти номер первой из строк, не содержащих ни одного положительного элемента.

Прикладные математические задачи

7. Найти все натуральные числа, не превосходящие заданного n , десятичная запись которых есть строго возрастающая или строго убывающая последовательность цифр.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Вариант 18

Линейные алгоритмы

1. Найти корни квадратного уравнения, заданного своими коэффициентами, с положительным дискриминантом; подстановкой в уравнение убедиться в погрешности вычислений.

Разветвляющиеся алгоритмы

2. Может ли шар радиуса R пройти через ромбообразное отверстие с диагоналями P и Q ?

Циклические вычислительные процессы

3. Вычислить и вывести на экран в виде таблицы значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX , функция определяется следующим образом:

$$F = \begin{cases} ax^3 + bx^2 & \text{при } x < 0 \text{ и } b \neq 0, \\ \frac{x-a}{x-c} & \text{при } x > 0 \text{ и } b = 0, \\ \frac{x+5}{c(x-10)} & \text{в остальных случаях,} \end{cases}$$

где a, b, c – действительные числа. Значения $a, b, c, X_{нач}, X_{кон}, dX$ вводятся с клавиатуры.

4. Числа $x_1, x_2, \dots$ последовательно поступают с устройства ввода. Все числа хранить в памяти нет необходимости; после ввода каждого числа нужно вычислить и напечатать среднее значение всех введенных чисел: $s_n = \frac{1}{n} \sum_{k=1}^n x_k$.

Простейшие операции с массивами

5. В одномерном массиве из N вещественных элементов, вычислить:

- количество элементов массива, меньших C ;
- сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального, не более чем на 20%, а потом – все остальные.

Двумерные массивы

6. Дана целочисленная прямоугольная матрица. Определить:

- количество строк, содержащих хотя бы один нулевой элемент;
- номер столбца, в которой находится самая длинная серия одинаковых элементов.

Прикладные математические задачи

7. Каждое из заданных натуральных чисел заменить числом, получающемся при записи его десятичных цифр в обратном порядке.

Указатели

В задачах 5 и 6 организовать данные с помощью указателей.

Работа с файлами

В задачах 5 – 7 организовать ввод данных из файла `input.txt` (в динамически организованные массивы в задачах 5 и 6 и вывод результата в файл `output.txt`).

Задачи для продолжающих

В этом разделе приведены варианты заданий по программированию задач по комбинаторике и теории вероятностей, а также задач, требующих использования визуальной среды C++ Builder.

Общие требования

Задача 1 разрабатывается как консольное приложение. Входные данные находятся в отдельном файле, все используемые массивы переменной длины – динамические, массивы, размеры которых указаны в условии задачи как постоянные – статические. Статический размер определяется директивой `#define`. Выходные данные записываются в файл, если в условии не оговорено иного.

Задачи 2–5 выполняются в виде Windows-приложений.

В задаче 2 необходимо вводить данные в компоненты `TEdit` таким образом, чтобы пользователь не имел возможности ввести некорректные данные (символьные строки вместо чисел; данные, не соответствующие условиям задачи).

В задаче 3 разработать приложение наподобие стандартного «Блокнота», позволяющее загружать данные из файла, имя которого указывается посредством диалогового окна `OpenDialog`, набирать текст в окне редактора, а также вставлять текст из буфера обмена. Приложение также должно обеспечивать запись результатов в файл, имя которого задается в диалоговом окне `SaveDialog`.

В задачах 4 и 5 изображение должно быть представлено непосредственно в рабочем окне. Если для работы программы необходим ввод параметров с клавиатуры – организовать область ввода данных, которая производит ввод в отдельном окне или же находится в нерабочей (не заполняемой рисунком) области окна (такую область выделяет автор программы). Прерывание работы программы должно выполняться по нажатию кнопки «Завершить», которую автор программы самостоятельно помещает на рабочую область.

Задачи считаются принятыми, если они прошли все тесты, введенные преподавателем, и студент ответил на возникшие у преподавателя вопросы.

Вариант 1

Комбинаторика и теория вероятностей

1. Матрицу $A(m, n)$ случайным образом заполнить разными целыми числами от 1 до $m \times n$.

Простейшие Windows-приложения

2. Написать Windows-приложение для вычисления стоимости покупки с учетом скидки 8%. Скидка предоставляется, если сумма покупки превышает 1000 рублей. Сумма покупки вводится в компонент TEdit, Информация о предоставлении скидки или отсутствии таковой выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. В древнеяпонском календаре был принят 60-летний цикл из пяти 12-летних циклов, обозначенных цветами: зеленый, красный, желтый, белый и черный. В подциклах каждый год имел название одного из животных: крысы, быка, тигра, кролика, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки и кабана. Началом очередного цикла был 4-ый год нашей эры, год зеленой крысы. Для заданного номера года найти его название по древнеяпонскому календарю.

Машинная графика

4. Заданное шестизначное десятичное натуральное число изобразить цифрами по 9-сегментному шаблону, используемому при почтовой индексации.

Элементы компьютерной анимации

5. Экран изображает собой бассейн с водой, в который бросили камень (в заданные координаты). От камня пошли круги, которые, дойдя до стенок бассейна, отражаются от них. Реализовать эту динамическую картину.

Вариант 2

Комбинаторика и теория вероятностей

1. Для заданных m и n найти все сочетания по m из чисел $1, 2, \dots, n$.

Простейшие Windows-приложения

2. Написать Windows-приложение для вычисления дохода по вкладу в банке. Доход вычисляется по формуле:

$$D = S \cdot (Sr/360) \cdot (St/100),$$

где S – сумма вклада, Sr – срок вклада (в днях), St – процентная ставка (в годовых). Параметры S , Sr , St вводятся пользователями в компонентах TEdit. Расчет процентов производится при нажатии кнопки. Полученный результат помещается в компоненту типа TLabel.

Алгоритмы обработки символьной информации

3. В текстовом файле представлены названия лекарств, срок их годности, количество упаковок и стоимость одной упаковки (через пробелы), например:

Аспирин 15.04.99 127 2.5
Парацетамол 27.09.01 23 5.40

Вывести на экран данные о лекарствах с истекшим сроком годности на сегодняшний день (с учетом перехода через столетие) и подсчитать общую стоимость таких медикаментов.

Машинная графика

4. Разработать орнамент на основе каких-либо математических кривых и заполнить ими окно.

Элементы компьютерной анимации

5. Изобразить на экране работающую ветряную мельницу. Плоскость вращения крыльев не обязательно параллельна плоскости экрана.

Вариант 3

Комбинаторика и теория вероятностей

1. Для заданного n получить все возможные перестановки чисел $1, 2, \dots, n$.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета стоимости поездки на автомобиле. С помощью компонент TEdit, пользователь вводит цену бензина (руб/литр), потребление бензина (литр/км) и расстояние (км). Информация о расходе бензина выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. В текстовом файле представлены в виде таблицы результаты соревнований по прыжкам в длину. В каждой строке файла записаны фамилия с инициалами спортсмена и через пробелы – его результаты в трех попытках. Найти трех призеров соревнования, не выделяя массива для хранения всех результатов.

Машинная графика

4. Получить на экране рисунок паутины с центром в произвольной (заданной) точке, с произвольным числом лучей. Паутина образована лучами и многоугольниками.

Элементы компьютерной анимации

5. Изобразить на экране работающие часы с 3 стрелками (часовая, минутная, секундная). Изменение времени – согласно таймеру.

Вариант 4

Комбинаторика и теория вероятностей

1. Перемешать массив $A(n)$ случайным образом так, чтобы ни один из элементов гарантированно не остался на своем прежнем месте и имел бы одинаковые

шансы занять любое из остальных мест. Разрешается использовать рабочий массив такой же размерности, что и основной.

Простейшие Windows-приложения

2. Написать Windows-приложение для перевода фунтов в килограммы. Данные (в фунтах) вводятся в компонент TEdit. Величина веса в килограммах выводится в компоненте типа TLabel после нажатия кнопки. Для справки: $kg = f \cdot 0.4995$.

Алгоритмы обработки символьной информации

3. Из имеющегося словаря (хранящегося в файле input.txt) выбрать наиболее длинное слово, в котором все буквы разные.

Машинная графика

4. Заданное число (не обязательно целое) отложить на бухгалтерских счетах, изображенных на экране.

Элементы компьютерной анимации

5. На экране – изображение бильярдного стола с одним шаром. Задается направление и начальная скорость после удара кием. Реализовать движение шара после удара до попадания его в лузу или до прерывания пользователем. Плоскость стола параллельна плоскости экрана, трением пренебречь.

Вариант 5

Комбинаторика и теория вероятностей

1. Имитировать перетасовку новой колоды игровых карт в 52 листа многократным применением операций сдвига и «врезки» так, чтобы никакие две рядом лежащие карты не сохранили бы свой первоначальный порядок. Операция врезки состоит в делении колоды на две подколоды и слиянии частей случайного объема, взятых из обеих подколод.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета силы тока в цепи, состоящей из соединенных последовательно 2 резисторов с сопротивлениями R_1 и R_2 . Данные вводятся в компоненты TEdit. Величина сопротивления выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Для заданного номера месяца вывести все приходящиеся на этот месяц праздничные дни. Например, 1 января – Новый год, 7 января – Рождество и т.д. Справочную информацию – список праздников – хранить в текстовом файле.

Машинная графика

4. Заданное число изобразить как на индикаторе микрокалькулятора.

Элементы компьютерной анимации

5. Изобразить движение футбольного мяча после удара (задается начальное положение мяча и вектор скорости). В процессе полета мяч ударяется о пол, потолок и стены, теряя при каждом ударе часть энергии.

Вариант 6

Комбинаторика и теория вероятностей

1. Из массива $A(n)$ удалить m наудачу (случайно) выбранных элементов. Порядок следования оставшихся элементов сохранить.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета силы тока в цепи, состоящей из соединенных параллельно 2 резисторов с сопротивлениями R_1 и R_2 . Данные вводятся в компоненты TEdit. Величина сопротивления выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Для заданного достаточно длинного слова найти в имеющемся словаре (хранящемся в файле input.txt) все слова, в которых использованы только буквы, имеющиеся в заданном слове (с учетом кратности вхождения).

Машинная графика

4. Изобразить на экране достаточно сложный цветок (георгин, ромашка со случайным числом лепестков, василек и т.п.).

Элементы компьютерной анимации

5. Частица (от заданной начальной точки) совершает хаотичное движение, двигаясь в случайном направлении на случайное расстояние (в пределах экрана). Получить на экране траекторию движения частицы до прерывания с клавиатуры (трением и потере энергии при ударах о границы пренебречь).

Вариант 7

Комбинаторика и теория вероятностей

1. Напечатать заготовку для игры «Морской бой», т.е. разложить случайным образом 15 «линейных» кораблей (1 – 5-клеточный, 2 – 4-клеточных, 3 – 3-клеточных, 4 – 2-клеточных, 5 – одноклеточных) на поле размером 10X10 так, чтобы они не касались друг друга.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета расхода электроэнергии. Данные (предыдущее и текущее показания счетчика, цена в рублях за 1 кВт) вводятся в компоненты TEdit. Сумма к оплате выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Примем следующие правила переноса русских слов:

- В каждой из разделяемых частей должно быть более одной буквы, из которых хотя бы одна – гласная;
- Нельзя разделять согласную и следующую за ней гласную;
- Буквы Й, Ъ, Ь считать согласными, но перенос после них допустим.

В каждом из вводимых слов поставить все возможные знаки переноса, например: СЕ-ЛЬ-С-КО-ХО-ЗЯЙ-С-Т-ВЕ-Н-НАЯ. Строчные и прописные буквы считать неразличимыми.

Машинная графика

4. Для заданного числа n построить двоичное дерево, содержащее n уровней.

Элементы компьютерной анимации

5. Изобразить действующий конвейер, транспортирующий какие-либо предметы.

Вариант 8

Комбинаторика и теория вероятностей

1. Заполнить квадратную матрицу $K(n, n)$ случайным образом числами $1, 2, \dots, n$ так, чтобы в каждой строке и в каждом столбце не было одинаковых чисел.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета площади треугольника, если известны его основание и высота (в см). Данные вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Один из способов идентификации автора литературного произведения – подсчет частоты вхождения отдельных слов. В заданном тексте (хранящемся в файле input.txt) найти 20 наиболее часто встречающихся слов с указанием количества использования каждого из них.

Машинная графика

4. Заполнить экран рисунком рыбьей чешуи с заданным размером элементарных чешуек.

Элементы компьютерной анимации

5. Экран – сосуд с кипящей жидкостью. На дне в случайной точке образуется пузырек, при движении вверх он растет, а дойдя до поверхности – лопается. Если два пузырька соприкасаются, они сливаются в один. Реализовать этот процесс.

Вариант 9

Комбинаторика и теория вероятностей

1. Расположить на шахматной доске случайным образом $m \leq 8$ слонов так, чтобы они не угрожали друг другу (слон бьет по диагонали).

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета стоимости покупки, состоящей из нескольких тетрадей и карандашей. Данные (количество предметов и цены за единицу продукции) вводятся в компоненты TEdit. Сумма к оплате выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. В имеющемся словаре (хранящемся в файле input.txt, например) найти группы слов, записанных одними и теми же буквами и отличающихся только их порядком, т.е. перестановкой. Например, КОМАР, КОРМА...

Машинная графика

4. Одинаковые трубы в количестве n штук уложены возможно более компактно – пирамидой. Получить на экране вид пирамиды с торца для произвольного числа n .

Элементы компьютерной анимации

5. В случайных точках местности расположены несколько столбов, некоторые из них соединены заборами. Пьяница с равной вероятностью делает шаг вперед, назад, вперед-вправо или вперед-влево (под 45 градусов). Натолкнувшись на столб или забор, он падает на некоторое время. Изобразить траекторию его движения.

Вариант 10

Комбинаторика и теория вероятностей

1. На плоскости заданы своими целочисленными координатами n точек. Найти всевозможные группы по 3, 4, ... точки, лежащие на одной прямой.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета стоимости некоторого количества (по весу) яблок. Данные (вес яблок и цена за килограмм) вводятся в компоненты TEdit. Сумма к оплате выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Каждую 80-байтную строку заданного в файле input.txt текста отцентрировать, т.е. обеспечить осевую симметрию текста на экране добавлением пробелов слева.

Машинная графика

4. Изобразить на экране всходы какого-либо растения, посаженного квадратно-гнездовым способом на грядке или на поле. Учесть перспективу.

Элементы компьютерной анимации

5. Изобразить в движении кубик заданного размера, равномерно вращающийся вокруг вертикально оси.

Вариант 11

Комбинаторика и теория вероятностей

1. Массив $X(n)$ разбит на m фрагментов. Длины фрагментов собраны в массиве $K(m)$. Переставить фрагменты случайным образом и преобразовать соответственно массивы X и K .

Простейшие Windows-приложения

2. Написать Windows-приложение для вычисления площади треугольника, если известны длины двух его сторон (см) и величина угла между ними (рад). Данные вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. В имеющемся в файле input.txt словаре найти пары слов, при прочтении каждого из которых в обратном направлении образуется другое слово пары, например, (ПОЛК, КЛОП), (БАР, РАБ).

Машинная графика

4. Изобразить на экране шахматную доску (вместе с буквенно-цифровыми обозначениями горизонталей и вертикалей) и случайным образом расставленные на ней шашки.

Элементы компьютерной анимации

5. Изобразить на экране звездное небо, полную луну и медленно надвигающуюся на нее тень Земли; затем медленное открытие диска Луны.

Вариант 12

Комбинаторика и теория вероятностей

1. Массив $Z(n)$ разбить на m фрагментов случайной длины. Фрагменты записать в матрицу A , содержащую m строк.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета стоимости поездки на автомобиле на дачу (туда и обратно). Данные (расстояние до дачи (км), количество бензина, которое потребляет автомобиль на 100 км пробега, цена одного литра

бензина в рублях) вводятся в компоненты TEdit. Результат вычислений выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. В одном файле задан набор ключевых слов, в другом файле дан длинный список названий книг и научных работ. Выбрать и записать в отдельный файл названия, содержащие хотя бы одно из заданных ключевых слов.

Машинная графика

4. Уравнение кривой $F(x, y) = 0$ не удалось представить в явном виде. Построить такую кривую.

Элементы компьютерной анимации

5. Изобразить на экране произвольный пейзаж, натюрморт или интерьер. Затем случайными точками или прямыми заполнить экран до полного исчезновения картины.

Вариант 13

Комбинаторика и теория вероятностей

1. Из массива $A(n)$ выбрать случайным образом m различных элементов и поместить их в массив $B(m)$.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета скорости, с которой спортсмен пробежал дистанцию. Данные (длина дистанции (м) и результат спортсмена (с)) вводятся в компоненты TEdit. Результат вычислений выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Текст, не содержащий собственных имен и сокращений, набран полностью прописными русскими буквами. Заменить все прописные буквы, кроме букв, стоящих после символов конца предложения (точка, «!», «?», «...» и т.д.), строчными буквами.

Машинная графика

4. Для некоторой функции $y = f(x, a)$ построить семейство графиков для различных значений параметра a .

Элементы компьютерной анимации

5. Получить на экране картину, которую видит машинист движущегося поезда: рельсы, шпалы, столбы, придорожные строения и т.д.

Вариант 14

Комбинаторика и теория вероятностей

1. «Перемешать» массив $T(m)$, переставив его элементы случайным образом. В массиве $K(m)$ запомнить позиции элементов до перемешивания.

Простейшие Windows-приложения

2. Написать Windows-приложение для пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах. Данные вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Заданный список русских фамилий (вместе с именами и отчествами) упорядочить по алфавиту. Проверить (и исправить, если нужно) написание собственных имен с прописных букв.

Машинная графика

4. Распределение скорости ветра по каждому из восьми направлений задано массивом из восьми чисел. Построить «розу ветров» с указанием направлений.

Элементы компьютерной анимации

5. Изобразить финальную сцену какого-либо театрального представления: на экране произвольное изображение; справа и слева на него надвигается занавес. На занавесе надпись: «КОНЕЦ».

Вариант 15

Комбинаторика и теория вероятностей

1. Задан массив $K(m)$ попарно различных целых чисел. Получить все перестановки целых чисел.

Простейшие Windows-приложения

2. Написать Windows-приложение, которое преобразует введенное с клавиатуры дробное число (например, 12.5) в денежный формат (12 руб. 50 коп.). Данные вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Имеется список членов коллектива с указанием принадлежности каждого к различным общественным организациям (профком, ученый совет, совет ветеранов, общество любителей животных и т.д.). Напечатать приглашения (в одном файле) всем членам на очередное заседание указанной организации. На входе задается только вид организации, место и время сбора.

Машинная графика

4. Изобразить произвольный домик (деревянный или кирпичный), отобразив материал стен (бревна, кирпичи). Размеры домика, кирпичей и окон задается вводом.

Элементы компьютерной анимации

5. Получить изображение движущегося математического маятника. Длина маятника и начальное положение – задаются пользователем.

Вариант 16

Комбинаторика и теория вероятностей

1. «Перемешать» матрицу $Z(m,n)$, переставив все ее элементы случайным образом.

Простейшие Windows-приложения

2. Написать Windows-приложение для расчета площади треугольника. Данные (координаты его углов на плоскости OXY) вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. По заданному списку фамилий напечатать каждому упомянутому в списке поздравление к определенному празднику. Чтобы избежать шаблона, перечень желаемых благ выбирать как случайное подмножество из заготовленного списка (например, здоровья, счастья, удачи, продвижения по службе, долголетия и т.д.).

Машинная графика

4. Построить мозаику из правильных шестиугольников заданного размера, закрасив соседние шестиугольники разными цветами.

Элементы компьютерной анимации

5. Разработать или реализовать динамическую рекламу какого-либо товара или услуги.

Вариант 17

Комбинаторика и теория вероятностей

1. Переставить все строки и столбцы матрицы $F(m,n)$ случайным образом. В массивах $K(m)$ и $L(n)$ запомнить места, которые каждая строка и каждый столбец занимали до перестановки.

Простейшие Windows-приложения

2. Написать Windows-приложение для вычисления площади кольца. Данные (радиусы кольца и отверстия) вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Стихотворный текст (в строке не более 80 символов) имеет четырех-строчную строфу. Записать его «лесенкой» (по одному слову в строке), вставляя пустую строку после каждого четверостишья.

Машинная графика

4. Заданный текст печатать крупными буквами с некоторыми паузами между буквами. Перенос осуществлять только полными словами.

Элементы компьютерной анимации

5. Изобразить модель атома произвольного химического элемента: ядро и электроны, вращающиеся по своим орбитам. Распределение электронов по орбитам задается.

Вариант 18

Комбинаторика и теория вероятностей

1. Стержень длиной l_0 нужно разрезать на стержни длиной $l_1, l_2, \dots, l_k$. Количество отрезков каждого типа не ограничено. Найти все способы раскроя и вычислить неиспользованный остаток в каждом способе.

Простейшие Windows-приложения

2. Написать Windows-приложение для пересчета времени, введенного в минутах и секундах в секунды. Данные вводятся в компоненты TEdit. Результат выводится в компоненте типа TLabel после нажатия кнопки.

Алгоритмы обработки символьной информации

3. Текст записан одной длинной строкой. Признаком красной строки служит символ \$. Переформатировать текст в 60-символьные строки, формируя абзацы.

Машинная графика

4. Рисунок на шляпке подсолнуха представляет собой семейство логарифмических спиралей, закрученных в разные стороны. Получить такой рисунок. Между прочим, количество «правых» и «левых» спиралей есть два соседних числа Фибоначчи.

Элементы компьютерной анимации

5. Разработать и реализовать на экране заставку к одной из популярных телепрограмм с движущимися эффектами.

Задания на курсовую работу

Цель курсовой работы – укрепление навыков разработки консольных и Windows-приложений путем решения двух достаточно объемных задач.

Задание 1. Общие положения

Windows-приложение должно иметь вид законченного программного продукта, готового к реализации. Кроме того, что эта программа должна выполнять описанные в условии функции, она должна иметь заставки и окна помощи, возможно, всплывающие подсказки. Базы данных необходимо создавать с использованием различных видов списков и структур данных. Данные из базы хранятся в текстовом файле. *В рамках данного задания не разрешается составлять базу с использованием стандартных компонент для работы с базами данных в C++ Builder.* Все вводимые данные должны проверяться. Если в условии задачи оговорена реализация удаление записей базы данных, необходимо при выполнении этой операции проверять целостность базы, т.е. не становится ли она противоречивой, нет ли ссылок на уже не существующие элементы. Записи в базах данных организовать в виде однонаправленных или двунаправленных списков (возможно, с головой). Вид списков выбирается автором каждой программы самостоятельно.

Отчет по заданию содержит постановку задачи, структуру базы данных, полную инструкцию для пользователя, описание алгоритма работы, экранных форм. Также прилагается полный текст файлов с расширениями *.cpr и *.h. Все формулы в отчетах вводятся с помощью встроенного редактора формул. При несоответствии отчета или программы данным требованиям, оценка за задание снижается.

Вариант 1. Личная библиотека. Картотека домашнее библиотеки: выходные данные книги (авторы, название, издательство и т.д.), раздел библиотеки (специальная литература, хобби, домашнее хозяйство, беллетристика и т.д.), происхождение и наличие книги в даны момент, субъективная оценка книги. В программе осуществить выбор книг по произвольному запросу, инвентаризацию библиотеки.

Вариант 2. Картотека Интерпола. Данные по каждому зарегистрированному преступнику: фамилия, имя, кличка, рост, цвет волос и глаз, особые приметы, гражданство, место и дата рождения, последнее место жительства, знание языков, преступная профессия, последнее дело и т.д. Преступные и мафиозные группировки, данные о подельщиках. Выборка по любому подмножеству признаков, Перенос «завязавших» в архив; удаление – только после смерти.

Вариант 3. Записная книжка. Анкетные данные, адреса, телефоны, место работы или учебы, должность знакомых, коллег и родственников, характер знакомства, деловые качества и так далее. В программе реализовать автоматическое формирование поздравлений с днем рождения (по текущее дате), упорядочивание

по алфавиту и по дате последней корректировки, поиск по произвольному шаблону.

Вариант 4. Касса аэрофлота. Расписание: номер рейса, маршрут, пункты промежуточной посадки, время отправления, дни полета. Программа должна определять количество свободных мест на каждом рейсе, выбирать ближайший рейс до заданного пункта (при наличии свободных мест), оформление заданного числа билетов по согласованию с пассажиром (с уменьшением числа свободных мест).

Вариант 5. Магазин с одним продавцом. Компьютер вместо кассового аппарата. База наличия товаров: наименование, единица измерения, цена единицы, количество, дата последнего завоза. Программа выполняет регистрацию поступления товаров (как старых, так и новых наименований) и оформление покупок: выписка чека, корректировка базы.

Вариант 6. Администратор гостиницы. Имеется список номеров: класс, число мест. Имеется список гостей: паспортные данные, даты приезда и отъезда, номер. Программа должна осуществлять поселение гостей: выбор подходящего номера (при наличии свободных мест), регистрация, оформление квитанции; оформлять документы при отъезде: выбор всех постояльцев, отъезжающих в конкретный день, освобождение мест и оформление задержки с выпиской дополнительной квитанции. Реализовать возможность досрочного отъезда с перерасчетом и поиск гостя по произвольному признаку.

Вариант 7. База данных меломана. База групп и исполнителей: база песен, дисков с перечнем песен (в виде ссылок), поиск песен заданной группы, всех дисков, где встречается заданная песня.

Вариант 8. Справочник врача. База болезней: название, симптомы, процедуры, перечень рекомендуемых лекарств с указанием требуемого количества. База медикаментов на складе: название, количество, взаимозаменяемость. Программа выполняет формирование рецепта после осмотра больного, проверку наличия лекарства, корректировку запасов.

Вариант 9. Обмен жилья. База предложений по обмену: район, площадь, планировка и т.д.; требования к Вариантам обмена. Программа осуществляет регистрацию клиентов, выбор подходящих Вариантов, удаление записей при состоявшемся обмене или отказе.

Вариант 10. Банк. Сведения о вкладчиках банка: номер лицевого счета, категория вкладчика, паспортные данные, текущая сумма вклада, дата последней операции. Программа осуществляет операции приема и выдачи любой суммы, автоматическое начисление процентов.

Вариант 11. Справочник коммерческих банков. База содержит наименование банка, его статус (форма собственности), условия хранения средств на лицевом счете (годовые проценты для различных видов вкладов). Программа осуществляет выбор банка с наибольшим процентом для заданного типа вклада.

Вариант 12. Справочник туриста. База содержит названия турагенств и предлагаемые ими услуги: страна, город (или маршрут круиза), условия проживания и проезда, экскурсионное обслуживание, сервис принимающей стороны,

стоимость путевки. Программа должна находить все подходящие под запросы клиента туры.

Вариант 13. Крылатые фразы. Справочник пословиц, поговорок, афоризмов, каламбуров, других словесных курьезов. Программа выполняет классификацию по авторам и источникам, поиск по темам и ключевым словам.

Вариант 14. Телепрограмма. Программа телепередач нескольких телекомпаний (на неделю по дням и часам). Разные жанры телепередач: новости, спорт, художественные фильмы, сериалы, мультфильмы и т.д. Программа должна осуществлять выбор совокупной программы по определенному запросу (вкусу) без временных накладок.

Вариант 15. Бюро знакомств. Имеются базы потенциальных женихов и невест: пол, регистрационный номер, дата регистрации, сведения о себе, требования к партнеру. Программа осуществляет выбор подмножества подходящих кандидатур, подготовка встреч (формирование приглашений для знакомства), перенос в архив пар, решивших свои семейные проблемы, отказавшихся от услуг.

Вариант 16. Ежедневник. База намеченных мероприятий – дата, время и протяженность, место проведения. Программа осуществляет автоматическое напоминание ближайшего дела: по текущее дате и времени; удаление вчерашних дел либо перенос на будущее, анализ «накладок» – пересечений планируемых дел, просмотр дел на завтра, послезавтра и т.д.

Вариант 17. Шеф-повар. Имеются: база данных рецептов блюд: раскладка, рецепт приготовления, и база данных продуктов на складе: наименование, цена, количество. Программа осуществляет формирование меню на день (на заданное число персон); званы ужин; проверку достаточности запасов; формирование расходной накладной на склад, корректировку запасов.

Вариант 18. Справочник абитуриента. Имеется база вузов: наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении. Программа осуществляет выбор по разным критериям: все о данном вузе, все о дано специальности, поиск минимального конкурса по данной специальности или вообще.

Задание 2

Сгенерировать одномерный массив из N случайных чисел $x_i \in [0;1]$. Отсортировать массив методом пузырька и методом A . Подсчитать число сравнений, при использовании каждого метода. Построить зависимость между числом элементов массива и числом операций сравнения ($N = 50, 100, 150, \dots, 1000$).

Под методом A следует понимать:

Вариант	Метод
1	Метод слияния
2	Метод простого обмена (шейкер-сортировка)

3	Сортировка Л. Шелла
4	Пирамидальная сортировка
5	Метод простого (прямого) выбора
6	Метод деления пополам (бинарные включения)
7	Метод простых вставок (простых включений)
8	Метод слияния
9	Метод простого обмена (шейкер-сортировка)
10	Сортировка Л. Шелла
11	Пирамидальная сортировка
12	Метод простого (прямого) выбора
13	Метод деления пополам (бинарные включения)
14	Метод простых вставок (простых включений)
15	Сортировка Л. Шелла
16	Пирамидальная сортировка
17	Метод простого (прямого) выбора
18	Метод деления пополам (бинарные включения)

Программа выполняется в виде консольного приложения. Результаты работы программы записываются в файл *.txt. График зависимостей строится с помощью стандартных средств *MS Office* по данным в выходном файле. Отчет по заданию должен содержать: постановку задачи, подробное описание методов сортировки, их алгоритмов, анализ сложности предложенных алгоритмов, результаты работы программы (график зависимости) и соответствующие выводы. В приложении привести тексты файлов *.cpp и *.h.

Дополнительные задачи (для продвинутых)

Ниже перечислены задачи из литературы разных лет.

Геометрия

1. Из заданного множества точек на плоскости выбрать две различные точки так, чтобы количества точек, лежащих по разные стороны прямой, проходящей через эти две точки, различались наименьшим образом.
2. Определить радиус и центр окружности, на которой лежит наибольшее число точек заданного на плоскости множества точек.
3. Задано множество M точек на плоскости. Определить, верно ли, что для каждой точки A из M существует точка B из M ($A \neq B$) такая, что не существует двух точек множества M , лежащих по разные стороны от прямой AB .
4. Определить радиус и центр такой окружности, проходящей хотя бы через три различные точки заданного множества точек на плоскости, что минимальна разность количеств точек, лежащих внутри и вне окружности.
5. В множестве точек на плоскости найти пару точек с максимальным расстоянием между ними.
6. Расстояние между двумя множествами точек – это расстояние между наиболее близко расположенными точками этих множеств. Найти расстояние между двумя заданными множествами точек на плоскости.
7. Многоугольник (не обязательно выпуклый) задан на плоскости перечислением координат вершин в порядке обхода его границы. Определить площадь многоугольника.
8. Задано множество точек M в трехмерном пространстве. Найти такую из них, что шар заданного радиуса с центром в этой точке содержит максимальное число точек из M .
9. Задано множество прямых на плоскости (коэффициентами своих уравнений). Подсчитать количество точек пересечения этих прямых.
10. В трехмерном пространстве задано множество материальных точек. Найти ту из них, которая наиболее близко расположена к центру тяжести этого множества.
11. В трехмерном пространстве задано множество материальных точек. Каждая из точек с максимальной массой исчезает, теряя десятую часть своей массы и раздавая оставшуюся массу поровну всем остальным, более «легким» точкам. Определить суммарную массу множества материальных точек в тот момент, когда все оставшиеся в нем точки имеют одинаковую массу.
12. Порядок на точках плоскости определим следующим образом: $(x, y) < (u, v)$, если либо $x < u$, либо $x = u$ и $y < v$. Перечислить точки заданного множества точек на плоскости в соответствии с этим порядком.
13. Заданы два множества точек на плоскости. Построить пересечение и разность этих множеств.

14. Множество точек на плоскости назовем регулярным, если вместе с каждой парой различных точек оно содержит также еще одну – третью – вершину правильного треугольника с вершинами в этих точках. Определить, регулярно ли заданное множество точек.

15. На плоскости задано n множеств по m точек в каждом. Среди точек первого множества найти такую, которая принадлежит наибольшему количеству множеств.

16. На плоскости заданы множество точек A и множество окружностей B . Найти две такие различные точки из A , что проходящая через них прямая пересекается с максимальным количеством окружностей из B .

17. На плоскости заданы множество точек A и множество прямых B . Найти две такие различные точки из A , что проходящая через них прямая параллельна наибольшему количеству прямых из B .

18. На плоскости заданы множество точек A и точка d вне его. Подсчитать количество (неупорядоченных) различных троек точек a, b, c из A таких, что четырехугольник $abcd$ является параллелограммом.

19. Определить радиус и центр окружности, проходящей по крайней мере через три различные точки заданного множества точек на плоскости и содержащей внутри наибольшее количество точек этого множества.

20. Выбрать три различных точки из заданного множества точек на плоскости так, чтобы была минимальной разность между количествами точек, лежащих внутри и вне треугольника с вершинами в выбранных точках.

21. Множество попарно различных плоскостей в трехмерном пространстве задано перечислением троек точек, через которые проходит каждая из плоскостей. Выбрать максимальное подмножество попарно непараллельных плоскостей.

22. Задано множество точек в трехмерном пространстве. Найти минимум радиусов шаров с центрами в этих точках, содержащих ровно n точек этого множества.

23. Выбрать три разные точки заданного на плоскости множества точек, составляющие треугольник наибольшего периметра.

24. Из заданного на плоскости множества точек выбрать такие три точки, не лежащие на одной прямой, которые составляют треугольник наименьшей площади.

25. Дано $3n$ точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество n треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга.

26. Задано множество точек на плоскости. Выбрать из них четыре разные точки, которые являются вершинами квадрата наибольшего периметра.

27. Из заданного множества точек на плоскости выбрать три разные точки A, B, C так, чтобы внутри треугольника ABC содержалось максимальное количество точек этого множества.

28. Из заданного множества точек на плоскости выбрать две различные точки так, чтобы окружности заданного радиуса с центрами в этих точках содержали внутри себя одинаковое количество заданных точек.

29. На плоскости заданы множество точек M и круг. Выбрать из M две различные точки так, чтобы наименьшим образом различались количества точек в круге, лежащие по разные стороны от прямой, проходящей через эти точки.

30. Даны два непересекающихся конечных множества точек на плоскости. Определить окружность, проходящую через k ($k > 3$) точек каждого из множеств.

31. Даны два множества точек на плоскости. Из первого множества выбрать три различные точки так, чтобы треугольник с вершинами в этих точках содержал (строго внутри себя) равное количество точек первого и второго множеств.

32. Даны два множества точек на плоскости. Найти центр и радиус окружности, проходящей через k ($k > 3$) точек первого множества и содержащей строго внутри себя m точек второго множества.

33. На плоскости задано множество попарно различных прямых (коэффициентами своих уравнений). Указать среди них ту прямую, которая имеет максимальное число пересечений с остальными прямыми.

34. Даны два множества точек на плоскости. Указать центр и радиус окружности, проходящей через k ($k > 3$) точек первого множества и содержащей строго внутри себя равное число точек первого и второго множеств.

35. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы треугольник с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь.

36. Даны два множества точек на плоскости. Выбрать четыре различные точки первого множества так, чтобы квадрат с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь.

37. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы круг, ограниченный окружностью, проходящей через эти три точки, содержал все точки второго множества и имел минимальную площадь.

38. Найти ромб наибольшей площади с вершинами в заданном множестве точек на плоскости.

39. Подсчитать количество равносторонних треугольников с различными длинами оснований и вершинами в заданном множестве точек на плоскости.

40. Задано множество точек на плоскости, не лежащих на одной прямой. Определить минимальное подмножество точек, после удаления которых остаются точки, лежащие на одной прямой.

41. Построить множество всех различных выпуклых четырехугольников с вершинами в заданном множестве точек на плоскости.

42. Построить множество всех различных остроугольных треугольников с вершинами в заданном множестве точек на плоскости.

43. На плоскости заданы множество точек и окружность радиусом R с центром в начале координат. Построить множество всех треугольников с вершинами в заданных точках, имеющих непустое пересечение с окружностью.

44. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной,

45. Среди треугольников с вершинами в заданном множестве точек на плоскости указать такой, стороны которого содержат максимальное число точек заданного множества.

46. Построить два треугольника с вершинами в заданном множестве точек на плоскости так, чтобы первый треугольник лежал строго внутри второго

47. На плоскости задано множество окружностей. Две окружности A и B назовем связанными, если они пересекаются либо существует третья окружность C заданного множества, связанная с A и B . Выбрать максимальное подмножество попарно не связанных друг с другом окружностей.

48. Задано множество точек на плоскости. Перечислить все различные максимальные подмножества точек, лежащих на одной прямой, которые содержат более двух точек.

49. Определить радиус и центр окружности минимального радиуса, проходящей хотя бы через три различные точки заданного множества точек на плоскости.

50. Найти такую точку заданного на плоскости множества точек, сумма расстояний от которой до остальных минимальна.

Матрицы, векторы

1. Найти максимальное из чисел, встречающихся в заданной матрице более одного раза.

2. Расстояние между k -й и l -й строками матрицы A определяется как

$$\sum_{j=1}^n |a_{kj} - a_{lj}|.$$

Указать номер строки максимально удаленной от первой строки заданной матрицы.

3. Для заданной перестановки A чисел $1, \dots, 100$ найти такое $k > 3$, при котором максимально

$$R(k) = \min_{1 \leq i \leq 100} |A^k(i) - i|.$$

4. Для заданной перестановки A чисел $1, \dots, 100$ найти максимальное k , при котором $A^k(i) = i$ для всех $i = 1, 2, \dots, 100$.

5. Задана перестановка A чисел $1, \dots, 100$. Для каждого $i = 1, 2, \dots, 100$ указать такое (зависящее от i) минимальное значение k , при котором $A^k(i) = i$.

6. Определить норму заданной матрицы A , т. е, число $\max_i \sum_j |a_{ij}|$.

7. Начиная с центра, обойти по спирали все элементы квадратной матрицы размером 13×13 (распечатывая их в порядке обхода).

8. Цепью вектора $A = A(1) A(2) \dots A(n)$ называется всякая последовательность индексов $i_1 i_2 \dots i_k$ такая, что $A(i_j) = i_{j+1}$. Построить максимальную по длине цепь заданного вектора.

9. По заданной квадратной матрице размером 10×10 построить вектор длиной 19, элементы которого – максимумы элементов, диагоналей, параллельных главной диагонали.

10. Две строки матрицы назовем *похожими*, если совпадают множества чисел, встречающихся в этих строках. Найти количество строк в максимальном множестве попарно непохожих строк заданной матрицы.

11. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

12. Построить целочисленную матрицу A размером 10×10 , элементы которой равны

$$a_{ij} = \begin{cases} C_i^j, & \text{при } i \geq j, \\ C_j^i, & \text{при } j \geq i, \end{cases}$$

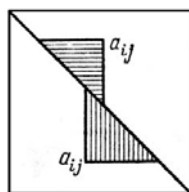
где C_l^k – число сочетаний из l элементов по k .

13. Найти номер строки заданной целочисленной матрицы размером 10×10 , в которой находится самая длинная серия одинаковых элементов.

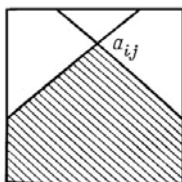
14. Для заданной целочисленной матрицы найти максимум среди сумм элементов на линиях, параллельных главной диагонали матрицы.

15. Для заданной целочисленной матрицы найти минимум среди сумм модулей элементов на линиях, параллельных побочной диагонали матрицы.

16. По матрице A размером 10×10 построить матрицу B того же размера, элемент b_{ij} которой равен минимальному элементу треугольника в A , определяемого элементом a_{ij} (см. рисунок).



17. По матрице A размером 10×10 построить матрицу B того же размера, где b_{ij} определяются следующим образом. Через a_{ij} проведем в A линии, параллельные главной и побочной диагоналям; b_{ij} определяется как максимум в заштрихованной части матрицы A (см. рисунок).



18. Дана матрица A размером 20×20 . Считая ее составленной из 100 квадратов размером 2×2 и переставляя эти квадраты, преобразовать A так, чтобы в результирующей матрице для всяких двух квадратов B к C выполнялось и следующее условие: если сумма элементов B меньше суммы элементов C , то B лежит либо выше, либо левее (когда B и C на одной горизонтали) квадрата C .

19. Говорят, что матрица имеет седловую точку a_{ij} , если a_{ij} является минимальным в i -й строке и максимальным в j -м столбце. Найти номер строки и столбца какой-нибудь седловой точки заданной матрицы.

20. Найти максимальный среди всех элементов тех строк заданной матрицы, которые упорядочены (либо по возрастанию, либо по убыванию).

21. Характеристикой столбца целочисленной! матрицы назовем сумму модулей его отрицательных нечетным элементов. Переставляя столбцы заданной матрицы, расположи их в соответствии с ростом характеристик.

22. Подсчитать количество столбцов заданной целочисленной матрицы размером 20×20 , которые составлены из попарно различных чисел.

23. Подсчитать количество строк заданной целочисленной матрицы размером 20×20 , являющихся перестановкой чисел $1, 2, \dots, 20$.

24. Напечатать элементы заданной матрицы размером 10×10 в следующем порядке:



25. Напечатать элементы заданной матрицы размером 10×10 в следующем порядке



26. Среди строк заданной целочисленной матрицы, содержащих только нечетные элементы, найти строку с максимальной суммой модулей элементов.

27. Среди столбцов заданной целочисленной матрицы, содержащих только такие элементы, которые по модулю не больше 10, найти столбец с минимальным произведением элементов.

28. Для заданной матрицы размером 10×10 найти такие k , что k -я строка матрицы совпадает с k -м столбцом.

29. Пусть $m(A, i)$ означает номер столбца матрицы A , в котором находится последний минимум i -й строки. Проверить, верно ли, что для заданной матрицы A размером 20×20 выполняются неравенства

$$m(A, 1) \leq m(A, 2) \leq \dots \leq m(A, 20).$$

30. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1$, $j-1 \leq l \leq j+1$, $(k, l) \neq (i, j)$. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 10×10 .

31. Всякий элемент a_{ij} заданной квадратной матрицы A размером 20×20 задает разбиение матрицы на четыре клетки с индексами

$$\{(k, l): 1 \leq k < i, 1 \leq l < j\}, \{(k, l): 1 \leq k < i, j \leq l \leq 20\}, \\ \{(k, l): i \leq k \leq 20, 1 \leq l < j\}, \{(k, l): i \leq k \leq 20, j \leq l \leq 20\},$$

хотя бы одна из которых непуста. Построить матрицу B того же размера, в которой b_{ij} равен минимуму среди максимальных элементов непустых клеток, определяемых в A элементом a_{ij} .

32. Определить, являются ли линейно независимыми три заданных вектора целых чисел длиной 30.

33. Проверить, удовлетворяет ли заданная матрица A размером 10×10 следующему условию: для всех $i > 1$ и для всех $j > 1$ верно неравенство $a_{ij} \geq a_{(i-1)j} + a_{i(j-1)}$

34. Для двух заданных матриц одинакового размера проверить, можно ли получить вторую матрицу из первой применением (конечного числа раз) операций транспонирования относительно главной и побочной диагоналей.

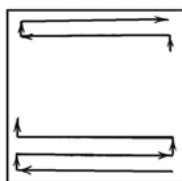
35. Найти и распечатать строку заданной целочисленной матрицы размером 10×10 , в которой длина максимальной серии одинаковых элементов минимальна.

36. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером 10×13 .

37. В условиях предыдущей задачи найти максимум среди всех локальных минимумов заданной матрицы размером 10×12 .

38. Определить, становится ли симметричной (относительно главной диагонали) заданная матрица размером 10×10 после замены на число 0 каждого локального минимума.

39. Напечатать элементы заданной матрицы размером 10×10 в следующем порядке:



40. Взаимно однозначное отображение элементов матрицы на себя можно задать с помощью двух целочисленных матриц: в первой указывать номер строки, куда переходит данный элемент, а во второй – номер столбца. Построить две матрицы, задающие отражение каждого элемента матрицы размером 10×10 на симметричный ему относительно главной диагонали.

Последовательности, тексты, предложения и слова

1. Проверить, имеется ли в заданном тексте баланс открывающих и закрывающих круглых скобок, т. е. верно ли, что можно установить взаимно однозначное соответствие открывающих и закрывающих скобок со следующими свойствами:

а) открывающая скобка всегда предшествует соответствующей закрывающей;

б) первый и последний символы текста – пара соответствующих друг другу скобок.

2. Для встречающихся в заданном тексте пар рядом расположенных символов указать, сколько раз встречается в тексте каждое из таких двухбуквенных сочетаний.

3. Для заданного текста определить длину содержащейся в нем максимальной серии символов, отличных от букв.

4. В заданной последовательности целых чисел найти самую длинную подпоследовательность, которая является арифметической или геометрической прогрессией.

5. По заданной последовательности целых чисел A построить последовательность B такую, что $B(i)$ – это количество элементов из A , превосходящих $A(i)$, в начальном отрезке A длиной $i - 1$.

6. Найти максимум длины таких начальных отрезков заданного слова, которые имеют вид vv , где v – симметричное слово.

7. Перечислить все слова заданного предложения, которые состоят из тех же букв, что и первое слово предложения.

8. Найти максимальную по длине монотонную (т. е. либо неубывающую, либо невозрастающую) подпоследовательность заданной последовательности целых чисел.

9. Для заданной последовательности целых чисел $A = A(1) A(2) \dots A(n)$ определим $T(i, j) = \sum_{k=i}^j A(k)$. Найти i, j такие, что $T(i, j)$ максимально.

10. Характеристикой слова назовем длину содержащейся в нем максимальной серии одинаковых символов. Упорядочить слова заданного предложения в соответствии с ростом их характеристик.

11. В заданном предложении найти пару слов, из которых одно является обращением другого.

12. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.

13. Найти самое длинное симметричное слово заданного предложения.

14. Расстояние между двумя словами равной длины – это количество позиций, в которых различаются эти слова. В заданном предложении найти пару наиболее далеко удаленных слов заданной длины.

15. Отредактировать заданное предложение, удаляя из него слова-серии, а также те слова, которые уже встречались в предложении раньше.

16. Из заданного текста выбрать и напечатать те символы, которые встречаются в нем ровно один раз (в том порядке, как они встречаются в тексте).

17. В предложении все слова начинаются с различных букв. Напечатать (если можно) слова предложения в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой следующего слова.

18. Найти множество всех слов, которые встречаются в каждом из двух заданных предложений.

19. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами.

Пример: НАША ТАНЯ ГРОМКО ПЛАЧЕТ -> ЯНАТ ТЕЧАЛП.

20. Заданы два текста; каждый текст составлен из попарно различных слов. Определить, можно ли получить второй текст из первого удалением некоторых его символов.

21. Найти самое длинное общее слово двух заданных предложений.

22. Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.

23. Среди слов заданного предложения, которые не являются сериями, найти такое, которое имеет наибольшее число вхождений в предложение.

24. Проверить, верно ли, что в заданном предложении всякое несимметричное слово имеет четную длину.

25. Отредактировать заданное предложение, удаляя из него слова, которые встречаются в предложении заданное число раз.

26. Определить, содержится ли в заданной последовательности целых чисел хотя бы одно число Фибоначчи.

27. Найти наименьшее общее кратное всех чисел, содержащихся в заданной последовательности натуральных чисел.

28. Проверить, является ли заданная последовательность целых чисел перестановкой начального отрезка последовательности натуральных чисел.

29. Напечатать заданное предложение таким образом, чтобы каждое его слово целиком находилось в одной и той же строке распечатки (т. е. избавиться от переносов).

30. Указать длину такого начального отрезка заданной последовательности целых чисел, для которого отношение встречающихся в нем степеней двойки и чисел Фибоначчи максимально.

31. В заданном предложении указать слово, в котором доля гласных (А, Е, И, О, У, Ы, Э, Ю, Я) максимальна.

32. Переставить и распечатать слова заданного предложения в соответствии с ростом доли согласных (Б, В, Г, Д, Ж, З, Й, К, Л, М, Н, П, Р, С, Т, Ф, Х, Ц, Ч, Ш, Щ, Ъ, Ь) в этих словах.

33. Для каждого символа заданного текста указать, сколько раз он встречается в тексте. Сообщение об одном символе должно печататься не более одного раза.

34. Отредактировать заданное предложение, удаляя из него все слова, целиком составленные из вхождений не более чем двух букв.

Пример: АННА ИВАНЕНКО -> ИВАНЕНКО.

35. Задан текст, в котором нет вхождений символов '(' и ')'. Выполнить его сжатие, т. е. заменить всякую максимальную подпоследовательность, составленную из более чем трех вхождений одного и того же символа, на $(k)s$, где s – повторяемый символ, а $k > 3$ – количество его повторений.

36. Отредактировать заданное предложение, заменяя всякое вхождение слова вида $\alpha\beta\bar{\alpha}$ на β , где α, β – полслова, $\bar{\alpha}$ – обращение слова α .

Задачи с целыми числами

1. Найти все такие простые числа, не превосходящие заданного N , двоичная запись которых представляет собой симметричную последовательность нулей и единиц (начинающуюся единицей!).

2. Построить таблицу всех различных разбиений заданного целого числа $N > 0$ на сумму трех натуральных слагаемых (разбиения, отличающиеся лишь порядком слагаемых, различными не считаются).

3. Для заданного натурального N определить наименьшее число S , которое можно представить в виде суммы $a^N + b^N$ по крайней мере двумя различными способами (a, b – натуральные числа; представления, отличающиеся лишь порядком слагаемых, различными не считаются).

4. Преобразовать заданное целое число из p -ичной системы счисления в q -ичную ($p, q \leq 16$; исходное число имеет не более n знаков).

5. Разложить заданное натуральное число на простые множители.

6. Построить первые N натуральных чисел, делителями которых являются только числа 2, 3 и 5.

7. Указать то число заданного множества целых чисел, в двоичном представлении которого больше всего единиц.

8. Перечислить все натуральные числа, не превосходящие заданного N , в двоичном представлении которых номера ненулевых разрядов образуют арифметическую прогрессию.

9. Среди простых чисел, не превосходящих N , найти такое, в двоичной записи которого максимальное число единиц.

10. Перечислить все пары «соседних» простых чисел, не превосходящих N , троичные представления которых получаются друг из друга записью цифр в обратном порядке (первая такая пара – это 5 и 7).

11. Найти все натуральные числа, не превосходящие заданного N и равные сумме кубов своих цифр.

12. Найти все пары двухзначных натуральных чисел M, N таких, что значение произведения $M \cdot N$ не изменится, если поменять местами цифры каждого из сомножителей (такой парой будет, например, 38 и 83).

13. Найти все натуральные числа, не превосходящие заданного N и делящиеся на каждую из своих цифр.

14. Заданы три натуральных числа A, B и N . Найти все натуральные числа, не превосходящие N , которые можно представить в виде суммы (произвольного числа) слагаемых, каждое из которых – A или B .

15. Найти все натуральные числа, не превосходящие заданного N , десятичная запись которых есть строго возрастающая или строго убывающая последовательность цифр.

16. Определить, можно ли представить заданное натуральное число как сумму кубов каких-нибудь трех натуральных чисел.

17. Определить, является ли периодической последовательностью двоичная запись заданного натурального числа N , т. е. имеет ли она вид $aa...a$, где a – некоторая непустая последовательность.

18. Найти все натуральные числа, не превосходящие заданного N и представимые в виде суммы квадратов двух каких-нибудь различных натуральных чисел.

19. Задано множество натуральных чисел. Заменить каждое из них на число, которое получается из исходного записью его (десятичных) цифр в обратном порядке. Полученное множество чисел распечатать.

20. Для натуральных A, B операцию $\odot$ определим так: $A \odot B = A - B + (A \% B)$. Найти все такие пары A, B , не превосходящие заданного N , для которых $A \odot B = B \odot A$.

21. Написать рекурсивную функцию для нахождения биномиальных коэффициентов, пользуясь их определением.

22. Задано конечное множество имен жителей некоего города, причем для каждого из жителей перечислены имена его детей. Жители X и Y называются родственниками, если

а) либо X – ребенок Y ,

б) либо Y – ребенок X ,

в) либо существует некий Z такой, что X является родственником Z , а Z является родственником Y .

Перечислить все пары жителей города, которые являются родственниками.

23. Подсчитать количество различных представлений заданного натурального N в виде суммы не менее двух попарно различных положительных слагаемых. Представления, отличающиеся лишь порядком слагаемых, различными не считаются.

Библиографический список

1. Касьянов, В.Н. Сборник заданий по практикуму на ЭВМ: уч. пос. для вузов / В.Н. Касьянов, В.К. Сабельфельд. – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 272 с.

2. Павловская, Т.А. С\С++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: Питер, 2002. – 464 с.: ил.

3. Семакин, И.Г. Основы программирования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2003. – 432 с.