

**Министерство образования и науки Российской Федерации
КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им.А.Н.Туполева**

Ю.В. Ефимова

Практикум по программированию на языке высокого уровня

Учебное пособие

Казань 2012

Министерство образования и науки Российской Федерации
КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им.А.Н.Туполева

Ю.В. Ефимова

Практикум по программированию на языке высокого уровня

Учебное пособие

*Рекомендовано к изданию Учебно-методическим центром
КГТУ им.А.Н.Туполева*

Казань 2012

Ю.В. Ефимова: Практикум по программированию на языке высокого уровня: Учебное пособие. Казань: Изд-во Казан. Гос. техн. Ун-та, 2012. 39 с.

ISBN

Представлены основные сведения и набор практических задач по разделам программирования, которые могут быть использованы и как варианты лабораторных работ по начальному курсу программирования на языке С. Пособие состоит из шести разделов, каждый из которых содержит теоретические сведения, примеры решения задач и упражнения для самостоятельного выполнения.

Предназначено для студентов направления вычислительной техники и информатики и других лиц, изучающих программирование. Может быть использовано для самоподготовки.

Ил. 7. Библиогр.: 6 назв.

Рецензенты: канд.техн.наук Зиновьев П.А.(Институт проблем информатики АН РТ)
канд.техн.наук Качалкин В.А. (Казанский институт(филиал) РГТЭУ)

ISBN

© Изд-во Казан. гос. техн. ун-та, 2012

© Ю.В.Ефимова , 2012

Введение

Развитие современных информационных технологий предполагает владение программистом широким набором практических навыков, среди которых одним из основных можно считать знание языка программирования и среды разработки.

Программирование - одно из наиболее важных направлений, способствующее формированию алгоритмического мышления учащихся. При ограниченном количестве часов преподавания необходимо применение рациональных методов обучения, часть из которых представлено в данном практикуме. Задачи по программированию удастся разбить на ряд классов и типизировать методы их решения.

1. Основные понятия программирования

Программирование - наука, изучающая теорию и методы разработки, производства и эксплуатации программного обеспечения ЭВМ.

Затраты на разработку, создание и эксплуатацию программ высоки и постоянно возрастают. До половины затрат на разработку программ требует ее отладка.

Отладка программы - обнаружение ошибок в программе, их локализация и исправление.

Методы отладки – тестирование, верификация.

Тестирование - выполнение программы вручную или на ЭВМ на контрольных примерах (тестах) с целью обнаружения ошибок или изучения механизма ее работы.

Тест - исходные данные программы вместе с ожидаемым правильным результатом работы.

Верификация - доказательство правильности программы в общем виде, по законам математики.

Алгоритм - это описание последовательности операций, направленной на решение поставленной задачи.

Основное свойство алгоритма – дискретность.

Процесс решения задачи представляется в виде последовательности шагов - операций. Операция - действие конечной продолжительности над некоторыми объектами. Операнд - объект, участвующий в операции.

Алгоритм состоит из операторов. Оператор - это описание операции.

Процесс – последовательность операций при выполнении алгоритма для конкретных исходных данных. Процессор - исполнитель алгоритма (процесса).

Программа – алгоритм для выполнения на ЭВМ.

Алгоритмический язык - система обозначений для записи алгоритмов.

Язык программирования - система обозначений для записи программ.

2. Программирование линейных алгоритмов на языке Си.

2.1.Способы записи алгоритма

1. Текстовый: в виде текста на естественном или искусственном языке, в том числе на языках программирования. Универсален, наиболее распространен.

2. Табличный: в форме двумерных таблиц (таблицы решений, таблицы переходов и т. п.) Не универсален, но удобен в отдельных случаях.

3. Графический. Наиболее нагляден. Используется в виде схем алгоритмов и программ.

Правила оформления схем регламентируются государственным стандартом ГОСТ 19.701-90. Схема состоит из блоков, обозначающих действия, соединенных стрелками, определяющими порядок выполнения действий. Основные символы, используемые в схемах алгоритмов представлены на рис 1.1.

Рис.1.1. Основные символы (виды блоков) схем алгоритмов

Блоки называются символами. Основными символами являются:

"процесс" (прямоугольник) - для описания операций ввода, обработки и вывода данных;

"решение" (ромб) - для разветвления алгоритма;

"пуск-останов" (овал) обозначает начало, конец и прерывание выполнения алгоритма;

"комментарий" - для пояснения схемы.

2.2.Пример

Даны два ненулевых числа. Найти их сумму и разность.

```
/* Программа 2.1.*/  
#include <stdio.h>  
main ()  
{  
float a, b,          /* числа a и b */  
s, r ;             /* сумма и разность */  
printf ("\n Введите два числа ");  
scanf ("%f %f", &a, &b);  
  
s = a + b;  
  
r = a - b;  
printf ("\n Сумма = %.2f\n ", s);  
printf ("\n Разность = %.2f\n ", r);  
return 0;  
}
```

Можно обойтись без дополнительных переменных s и r, тогда вычисление суммы и разности выполняется при выводе результата:

```
printf ("\n Сумма =%.2f\n ", a + b);  
и printf ("\n Разность =%.2f\n ", a - b);
```

Пояснения к программе.

1. /* */ - комментарий языка C не влияет на работу программы.

2. main() – заголовок главной функции, круглые скобки после имени main указывают на то, что это функция. Тело функции заключено в фигурные скобки {...}.

3. Программа на языке C состоит из одной или нескольких функций, выполнение программы начинается с функции main().

4. float a, b, s, r; – объявлены переменные вещественного типа.

5. printf () - вызов стандартной функции для вывода сообщения и результата. В скобках задаются параметры вызова - форматная строка в кавычках и возможно список выводимых значений.

```
printf ("\n Введите два числа ");
```

Форматная строка содержит текст в кавычках, где символ \n - переход на новую строку.

```
printf ("\n Сумма = %.2f\n ", s);
```

Форматная строка, кроме текста, содержит формат вывода переменной s. Выводится значение переменной s или выражения a+b.

%f – формат ввода/вывода вещественного числа,

%.2f - точность вывода до 2-х знаков после запятой.

6. scanf() - вызов стандартной функции для ввода исходных данных.

Параметры вызова - форматная строка в кавычках и имена переменных. Символ & перед именем переменной означает операцию получения адреса переменной.

7. #include <stdio.h> - директива препроцессора для использования стандартных функции ввода/вывода.

<stdio.h> - заголовочный файл, содержит описания стандартных функций ввода/вывода.

8. return - оператор возврата, завершает выполнение функции, return 0; означает успешное завершение программы.

2.3 Задачи.

1. Даны два ненулевых числа. Найти их произведение и частное.
2. Даны два числа. Найти среднее арифметическое их квадратов и среднее арифметическое их модулей.
3. Скорость лодки в стоячей воде V км/ч, скорость течения реки U км/ч (скорость течения реки меньше скорости лодки в стоячей воде). Время движения лодки по озеру T_1 ч, а по реке (против течения) — T_2 ч. Определить путь S , пройденный лодкой.
4. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили удаляются друг от друга.
5. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили первоначально движутся навстречу друг другу.
6. Найти периметр и площадь прямоугольного треугольника, если даны длины его катетов a и b .
7. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
8. Найти длину окружности и площадь круга заданного радиуса R . В качестве значения Pi использовать 3.14.
9. Найти площадь кольца, внутренний радиус которого равен R_1 , а внешний радиус равен R_2 . В качестве значения Pi использовать 3.14.
10. Дана сторона равностороннего треугольника. Найти площадь этого треугольника и радиусы вписанной и описанной окружностей.

11. Дана длина окружности. Найти площадь круга, ограниченного этой окружностью. В качестве значения Pi использовать 3.14.
12. Дана площадь круга. Найти длину окружности, ограничивающей этот круг. В качестве значения Pi использовать 3.14.
13. Найти периметр и площадь равнобедренной трапеции с основаниями a и b ($a > b$) и углом $alpha$ при большем основании (угол дан в радианах).
14. Найти периметр и площадь прямоугольной трапеции с основаниями a и b ($a > b$) и острым углом $alpha$ (угол дан в радианах).
15. Найти расстояние между двумя точками с заданными координатами $(x1, y1)$ и $(x2, y2)$.
16. Даны координаты трех вершин треугольника $(x1, y1)$, $(x2, y2)$, $(x3, y3)$.
Найти его периметр и площадь.
17. Найти корни квадратного уравнения $A \cdot x^2 + B \cdot x + C = 0$, заданного своими коэффициентами A, B, C (коэффициент A не равен 0), если известно, что дискриминант уравнения неотрицателен.
18. Составить программу перевода радианной меры угла в градусы, минуты и секунды.
19. Полторы кошки за полтора часа съедают полторы мышки. Сколько мышек съедят X кошек за Y часов.
20. Составить программу вычисления объема цилиндра, с высотой H и радиусом основания R .

3. Операторы ветвления

3.1.Оператор if

Условный оператор *if* используется для разветвления процесса вычисления на два направления .

Условный оператор *if* имеет вид:

if (выражение) оператор_1; [else оператор_2;]

Сокращенный условный оператор:

if (выражение) оператор 1;

Выполнение оператора *if* начинается с вычисления выражения.

Далее выполнение осуществляется по следующей схеме:

- если *выражение* истинно(т.е. $\neq 0$), то выполняется *оператор_1*.
- если *выражение* ложно (т.е. равно 0),то выполняется *оператор_2*.

Схема работы полного оператора *if* представлена на рис.2.1.

Рис.3.1. Схема работы полного оператора if

Схема работы сокращенного оператора *if* представлена на рис.2.2.

Рис.3.2. Схема работы сокращенного оператора *if*

3.2.Оператор *switch*

Оператор *switch* предназначен для разветвления процесса вычисления на несколько направлений.

Оператор *switch* имеет вид:

switch (*выражение*)

```
{  
[ case константное_выражение_1]: [ список_операторов_1]  
[ case константное_выражение_2]: [ список_операторов_2]  
...  
[ case константное_выражение_n [ список_операторов_n]  
[ default: [ список_операторов ]]  
}
```

Схема выполнения оператора *switch* следующая:

- вычисляется выражение в круглых скобках;
- вычисленные значения последовательно сравниваются с константными выражениями, следующими за ключевыми словами *case*;

- если одно из константных выражений совпадает со значением выражения, то управление передается на оператор, помеченный соответствующим ключевым словом *case*;

- если ни одно из константных выражений не равно выражению, то управление передается на оператор, помеченный ключевым словом *default*, а в случае его отсутствия управление передается на следующий после *switch* оператор.

Схема работы оператора *switch* представлена на рис.2.3.

Рис.3.3. Структурная схема оператора *switch*

3.3.Примеры

Даны два ненулевых числа. Вычислить квадрат наибольшего из них.

```
/* Программа 3.1.*/  
#include <stdio.h>  
main ()  
{  
float a, b,          /* числа a и b */  
s;                  /* квадрат наибольшего */  
printf ("\n Введите два числа ");  
scanf ("%f %f", &a, &b);
```

```

if (a>b) s = a*a;

else s = b*b;

printf ("\n Квадрат наибольшего = %.2f\n ", s);

return 0;

}

```

Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа A и B (B не равно нулю). Выполнить над числами указанное действие и вывести результат.

```

/* Программа 3.2.*/
#include <stdio.h>

main ()
{
float a, b,          /* числа a и b */
x;                 /* результат вычисления*/
int nomer;         /* номер действия */
printf ("\n Введите два числа ");
scanf ("%f %f", &a, &b);
printf ("\n Введите действие ");
scanf ("%d", &nomer);
switch (nomer)
{
case 1: x = a + b; break;
case 2: x = a - b; break;
case 3: x = a * b; break;
case 4: x = a / b; break;
default : ;
}
}

```

```
printf ("\n Результат операции = %.2f \n ", x);  
}
```

3.4. Задачи

1. Даны три целых числа. Возвести в квадрат отрицательные числа и в третью степень — положительные (число 0 не изменять).
2. Из трех данных чисел выбрать наименьшее.
3. Из трех данных чисел выбрать наибольшее.
4. Из трех данных чисел выбрать наименьшее и наибольшее.
5. Перераспределить значения переменных X и Y так, чтобы в X оказалось меньшее из этих значений, а в Y — большее.
6. Значения переменных X , Y , Z поменять местами так, чтобы они оказались упорядоченными по возрастанию.
7. Значения переменных X , Y , Z поменять местами так, чтобы они оказались упорядоченными по убыванию.
8. Даны две переменные целого типа: A и B . Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.
9. Даны две переменные целого типа: A и B . Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения.
10. Даны три переменные: X , Y , Z . Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

11. Даны три переменные: X , Y , Z . Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.
12. Даны целочисленные координаты точки на плоскости. Если точка не лежит на координатных осях, то вывести 0. Если точка совпадает с началом координат, то вывести 1. Если точка не совпадает с началом координат, но лежит на оси OX или OY , то вывести соответственно 2 или 3.
13. Даны вещественные координаты точки, не лежащей на координатных осях OX и OY . Вывести номер координатной четверти, в которой находится данная точка.
14. На числовой оси расположены три точки: A , B , C . Определить, какая из двух последних точек (B или C) расположена ближе к A , и вывести эту точку и ее расстояние от точки A .
15. Даны три положительных числа A, B, C . Проверить будут ли они сторонами треугольника? Если да, то вычислить площадь этого треугольника.
16. Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Вывести порядковый номер этого числа.
17. Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.
18. Дан номер некоторого года (положительное целое число). Вывести число дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).
19. Для данного x вычислить значение следующей функции $f(x)$, принимающей

вещественные значения:

$$f(x) = \begin{cases} x \cdot x, x > 0 \\ x + x, x \leq 0 \end{cases}$$

20. Для данного x вычислить значение следующей функции $f(x)$, принимающей вещественные значения:

$$f(x) = \begin{cases} 2 + x^3, x > 0 \\ x + x^2, x \leq 0 \end{cases}$$

21. Для данного x вычислить значение следующей функции $f(x)$, принимающей вещественные значения:

$$f(x) = \begin{cases} -1, x \leq 0 \\ x \cdot x, 0 < x < 2 \\ 4, x \geq 2 \end{cases}$$

22. Для данного x вычислить значение следующей функции $f(x)$, принимающей значения целого типа:

$$f(x) = \begin{cases} 0, x < 0 \\ 1, x \in [0,1), [2,3), \dots \\ -1, x \in [1,2), [3,4), \dots \end{cases}$$

23. Дано целое число, лежащее в диапазоне от -999 до 999 . Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.

24. Дано целое число, лежащее в диапазоне от 1 до 9999 . Вывести строку — словесное описание данного числа вида "четное двузначное число", "нечетное четырехзначное число" и т.д.

25. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.).

26. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней в этом

месяце для невисокосного года.

27. Дано целое число в диапазоне 0 – 9. Вывести строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два",...).
28. Дано целое число в диапазоне 1 – 5. Вывести строку — словесное описание соответствующей оценки (1 — "плохо", 2 — "неудовлетворительно", 3 — "удовлетворительно", 4 — "хорошо", 5 — "отлично").
29. Элементы окружности пронумерованы следующим образом: 1 — радиус (R), 2 — диаметр (D), 3 — длина (L), 4 — площадь круга (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности. В качестве значения Pi использовать 3.14.
30. Дано целое число в диапазоне 20 – 69, определяющее возраст (в годах). Вывести строку — словесное описание указанного возраста, обеспечив правильное согласование числа со словом "год", например: 20 — "двадцать лет", 32 — "тридцать два года", 41 — "сорок один год".

4. Операторы цикла

Цикл используется для организации многократно повторяющихся действий над данными. Циклы позволяют описывать длинные последовательности операций в виде коротких алгоритмов.

Один проход цикла называется итерацией. Проверка условия выполняется на каждой итерации либо до тела цикла (тогда говорят о цикле с предусловием), либо после тела цикла (цикл с постусловием).

4.1. Цикл с параметром *for*

Оператор цикла *for* имеет вид:

```
for ([выражение_1]; [выражение_2]; [выражение_3])  
  { оператор_1; оператор_2; }
```

Схема работы оператора *for* представлена на рис.4.1.

Рис. 4.1. Схема работы оператора *for*

4.2. Цикл с предусловием *while*

Оператор *while* – оператор цикла с предусловием.

Оператор цикла *while* имеет вид:

while (выражение)
оператор_1;

Оператор_1 повторяется ≥ 0 раз, пока истинно условие цикла, т.е. значение выражение $\neq 0$. Если условие сразу ложно, то оператор не выполнится ни разу.

Схема работы оператора *while* представлена на рис.4.2.

Рис.4.2. Схема работы оператора цикла *while*

4.3. Цикл с постусловием *do while*

Оператор цикла с постусловием имеет вид:

do
оператор_1;
while (выражение);

Сначала выполняется простой или составной *оператор_1*, составляющий

тело цикла, а затем вычисляется *выражение*. Если условие цикла истинно, т.е. значение *выражение* $\neq 0$, то тело цикла выполняется еще раз. Цикл завершится, когда *выражение* станет ложно.

Схема работы оператора *do while* представлена на рис.4.3.

Рис.4.3. Схема работы оператора *do while*

4.4.Пример

Даны числа N, A_1, A_2, \dots, A_N . Вычислить произведение числовой последовательности.

```
/* Программа 4.1. */
#include <stdio.h>
void main(void)
{ int n;                /* Количество чисел      */
  float x,              /* Текущее число      */
        pr=1;          /* Произведение       */
  int i;
  scanf("%d", &n);
  for ( i = 1; i <=n; i++)
    { scanf("%f", &x);    /* Ввод текущего числа */
```

```

    pr *= x;
}
printf ("\nПроизведение равно %.2f\n", pr);
}

```

4.5. Задачи

1. Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания, а также количество N этих чисел.
2. Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (не включая сами эти числа), в порядке их убывания, а также количество N этих чисел.
3. Дано вещественное число A и целое число N ($N > 0$). Вывести A в степени N : $A^N = A \cdot A \cdot A \cdot \dots \cdot A$ (числа A перемножаются раз).
4. Дано вещественное число A и целое число N ($N > 0$). Вывести все целые степени числа A от 1 до N .
5. Дано вещественное число A и целое число N ($N > 0$). Вывести $1 + A + A^2 + A^3 + \dots + A^N$.
6. Дано вещественное число A и целое число N ($N > 0$). Вывести $1 - A + A^2 - A^3 + \dots + (-1)^N \cdot A^N$.
7. Дано целое число N (). Вывести наименьшее целое K , при котором выполняется неравенство , и само значение .
8. Дано целое число N (). Вывести наибольшее целое K , при котором

выполняется неравенство $3^k < N$, и само значение 3^k .

9. Дано вещественное число A ($A > 1$). Вывести наименьшее из целых

чисел N , для которых сумма $1 + \frac{1}{2} + \dots + \frac{1}{N}$ будет больше A , и саму эту сумму.

10. Дано вещественное число A ($A > 1$). Вывести наибольшее из целых

чисел N , для которых сумма $1 + \frac{1}{2} + \dots + \frac{1}{N}$ будет меньше A , и саму эту сумму.

11. Дано целое число N ($N > 0$). Вывести произведение $1 \cdot 2 \cdot \dots \cdot N$. Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.

12. Дано целое число N ($N > 0$). Если N — нечетное, то вывести произведение $1 \cdot 3 \cdot \dots \cdot N$; если N — четное, то вывести произведение $2 \cdot 4 \cdot \dots \cdot N$. Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.

13. Дано целое число N ($N > 0$). Вывести сумму

$2 + \frac{1}{(2!)} + \frac{1}{(3!)} + \dots + \frac{1}{(N!)}$ (выражение $N!$ — "N факториал" — обозначает

произведение всех целых чисел от 1 до N : $N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением константы $e = \exp(1)$ ($= 2.71828183\dots$).

14. Дано вещественное число X и целое число N (). Вывести

().

15. Дано вещественное число X и целое число N (). Вывести

$$X - \frac{X^3}{3!} + \frac{X^5}{5!} - \dots + (-1)^N \frac{X^{2 \cdot N + 1}}{(2 \cdot N + 1)!}, \quad (N! = 1 \cdot 2 \cdot \dots \cdot N).$$

Полученное число является приближенным значением функции $\sin(x)$ в точке X .

16. Дано вещественное число X и целое число N ($N > 0$). Вывести

$$1 - \frac{X^2}{2!} + \frac{X^4}{4!} - \dots + (-1)^N \frac{X^{2 \cdot N}}{(2 \cdot N)!}, \quad (N! = 1 \cdot 2 \cdot \dots \cdot N).$$

Полученное число является приближенным значением функции $\cos(x)$ в точке X .

17. Дано вещественное число X ($|X| < 1$) и целое число N ($N > 0$). Вывести

$$X - \frac{X^2}{2} + \frac{X^3}{3} - \dots + (-1)^{N-1} \frac{X^N}{N}.$$

18. Дано вещественное число X ($|X| < 1$) и целое число N ($N > 0$). Вывести

$$X - \frac{X^3}{3} + \frac{X^5}{5} - \dots + (-1)^N \frac{X^{2 \cdot N + 1}}{(2 \cdot N + 1)}$$

Полученное число является приближенным значением функции $\arctg(x)$ в точке X .

19. Дано число D ($D > 0$). Последовательность чисел A_N определяется следующим образом:

$$A_1 = 2, A_N = 2 + \frac{1}{A_{N-1}}, N = 2, 3, \dots$$

Найти первый из номеров K , для которых

выполняется условие $|A_K - A_{K-1}| < D$, и вывести этот номер, а также числа A_{K-1} и A_K .

20. Дано число D (). Последовательность чисел определяется следующим образом:

Найти первый из номеров K , для которых выполняется условие ,

и вывести этот номер, а также числа A_{k-1} и A_k .

5. Одномерные массивы

Массив - упорядоченная последовательность пронумерованных элементов одинакового типа.

Индекс - номер элемента массива. Индексы элементов массива начинаются с нуля.

Элемент массива может иметь несколько индексов. Размерность массива - количество индексов каждого элемента.

Вектор - одномерный массив (один индекс).

Определение массива:

```
char m[10];          /* символьный вектор m */
float r[20];        /* вещественный вектор r */
```

5.1. Пример

Дан массив размера N. Вывести среднее арифметическое его элементов.

```
/*Программа 5.1 */
#include <stdio.h>
#define MAX 100 /* максимальное количество элементов в массиве */
void main()
{
float s=0; /* сумма элементов массива */
float sr_ar; /* среднее арифметическое элементов массива */
float mass[MAX];
int n=0; /* количество элементов в массиве*/
int i; /* счетчик */
printf ("\nВведите количество элементов в массиве: ");
scanf ("%d",&n);
printf ("Введите элементы массива \n");
for (i=1; i<=n; i++)
{
```

```

scanf ("%f",&mass[i]);
s+=mass[i];
}
sr_ar=s/n;
printf("Среднее арифметическое элементов массива =: %5.2f ",sr_ar);
}

```

5.2.Задачи

1. Дан массив размера N . Вывести его элементы в обратном порядке.
2. Дан массив размера N . Вывести вначале его элементы с четными индексами, а затем — с нечетными.
3. Дан целочисленный массив A размера 10. Вывести номер первого из тех его элементов $A[i]$, которые удовлетворяют двойному неравенству:
 $A[1] < A[i] < A[10]$. Если таких элементов нет, то вывести 0.
4. Дан целочисленный массив размера N . Преобразовать его, прибавив к нечетным числам последний элемент. Последний элемент массива не изменять.
5. Дан целочисленный массив размера N . Преобразовать его, прибавив к четным числам первый элемент. Первый элемент массива не изменять.
6. Дан целочисленный массив размера N . Вывести вначале все его четные элементы, а затем — нечетные.
7. Поменять местами минимальный и максимальный элементы массива размера 10. Заменить все положительные элементы целочисленного массива размера 10 на значение максимального.
8. Дан массив размера 10. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.

9. Дан массив размера N . Осуществить циклический сдвиг элементов массива влево на одну позицию.
10. Дан массив размера N . Осуществить циклический сдвиг элементов массива вправо на одну позицию.
11. Дан массив размера N и число k ($0 < k < 5$, $k < N$). Осуществить циклический сдвиг элементов массива влево на k позиций. Проверить, образуют ли элементы целочисленного массива размера N арифметическую прогрессию. Если да, то вывести 1, если нет — вывести 0.
12. Проверить, образуют ли элементы целочисленного массива размера N геометрическую прогрессию. Если да, то вывести 1, если нет — вывести 0.
13. Дан массив ненулевых целых чисел размера N . Проверить, чередуются ли в нем положительные и отрицательные числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
14. Дан массив ненулевых целых чисел размера N . Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
15. Дан массив размера N . Определить количество участков, на которых его элементы монотонно убывают.
16. Дан массив размера N . Определить количество участков, на которых его элементы монотонно возрастают.
17. Дан массив размера N . Определить количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).
18. Дан целочисленный массив размера N . Определить максимальное количество его одинаковых элементов.

19. Дан целочисленный массив размера N . Если он является перестановкой, то есть содержит все числа от 1 до N , то вывести 0, в противном случае вывести номер первого недопустимого элемента.
20. Дан массив размера N . Преобразовать его, вставив после каждого положительного элемента нулевой элемент.
21. Даны два массива A и B из 5 элементов каждый, элементы которых упорядочены по возрастанию. Объединить эти массивы так, чтобы результирующий массив остался упорядоченным.
22. Упорядочить массив размера N по убыванию.
23. Упорядочить массив размера N по возрастанию.
24. Дано множество A из N точек. Найти пару различных точек этого множества с максимальным расстоянием между ними и само это расстояние (точки выводятся в том же порядке, в котором они перечислены при задании множества A).
25. Дано множество A из N точек. Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.
26. Дано множество A из N точек с целочисленными координатами. Порядок на координатной плоскости определим следующим образом: $(x_1, y_1) < (x_2, y_2)$, если либо $x_1 < x_2$, либо $x_1 = x_2$ и $y_1 < y_2$. Расположить точки данного множества по возрастанию в соответствии с указанным порядком.
27. Дано множество A из N точек с целочисленными координатами. Порядок на координатной плоскости определим следующим образом: $(x_1, y_1) < (x_2, y_2)$, если либо $x_1 < x_2$, либо $x_1 = x_2$ и $y_1 < y_2$. Расположить точки данного множества по убыванию в соответствии с указанным порядком.

6. Двумерные массивы

Многомерные массивы в С рассматриваются как массивы, элементами которых являются массивы. Определение многомерного массива должно содержать информацию о типе, размерности и количестве элементов каждой размерности.

Например, оператор `int matr[5][8];` задает описание двумерного массива из 5 строк и 8 столбцов.

Для доступа к элементу массива указываются все его индексы, например, `matr[i][j]`.

6.1. Пример

Дана квадратная матрица вида

$$\begin{pmatrix} 8 & -11 & 12 & -34 & 29 \\ 18 & -41 & -42 & 82 & 35 \\ 36 & 4 & -2 & 5 & -18 \\ 48 & -36 & 30 & 41 & -28 \\ -32 & -23 & 3 & 1 & 2 \end{pmatrix}$$

Найти произведение положительных и отрицательных элементов каждой строки.

```
/* Программа 6.1*/
#include<stdio.h>
void main ()
{
long prp=1,      /* произведение положительных элементов*/
protr=1;        /* произведение отрицательных элементов*/
int i,j;
const int nstr =5,nstb=5;      /* размерность массива*/
int mas[nstr][nstb] ={        /* инициализация массива*/
{8,-11,12,-34,29},
```

```

{18,-41,-42,82,35},
{36,4,-2,5,-18},
{48,-36,30,41,-28},
{-32,-23,3,1,2}};
for(i=0;i<nstr; i++)          /* просмотр массива по строкам*/
{
    for(j = 0;j<nstb;j++)
        {
            if(mas[i][j]>0)
                prp=prp*mas[i][j];
            else
                protr=protr*mas[i][j];
        }
    printf("\nПроизведение положительных элементов %d строки равно
%d",i,prp);
    prp=1;
    printf("\nПроизведение отрицательных элементов %d строки равно
%d",i,protr);
    protr=1;
}
}

```

6.2.Задачи

1. В прямоугольной матрице, размерностью $n \times m$, найти сумму и количество элементов, попадающих в промежуток от A до B .

2. В прямоугольной матрице, размерностью $n \times m$, количество элементов, не попадающих в промежуток от A до B .

3. В прямоугольной матрице, размерностью $n \times m$, найти сумму и количество положительных элементов, которые лежат выше главной диагонали.

4. В прямоугольной матрице, размерностью $n * m$ найти сумму и количество элементов, которые являются простыми числами.

5. В прямоугольной матрице, размерностью $n * m$, найти сумму элементов в строках с $k1$ -й по $k2$ -ю.

6. В прямоугольной матрице, размерностью $n * m$, найти номера всех максимальных элементов.

7. В прямоугольной матрице, размерностью $n * m$, найти номера первых отрицательных элементов каждой строки (столбца).

8. В прямоугольной матрице, размерностью $n * m$, найти номера последних отрицательных элементов каждой строки (столбца);

9. В прямоугольной матрице, размерностью $n * m$, найти количество элементов в каждой строке, больших (меньших) среднего арифметического элементов заданной строки.

10. Определить, есть ли в данном двумерном массиве два одинаковых элемента.

11. Определить, есть ли в данном двумерном массиве данное число A среди элементов массива.

12. Определить является ли двумерный массив логическим квадратом, то есть суммы по всем горизонталям, вертикалям и двум диагоналям должны быть равны.

13. Определить является ли двумерный массив логическим квадратом, то есть суммы по всем горизонталям, вертикалям и двум диагоналям должны быть равны и сумма должна быть равна данному числу A .

14. Определить, есть ли в данном массиве строка, состоящая только из положительных элементов.

15. Определить, есть ли в данном массиве столбец, состоящий только из положительных элементов.

16. Определить, есть ли в данном массиве столбец, состоящий только из нулевых элементов.

17. Определить, есть ли в данном массиве строка, состоящая только из нулевых элементов.
18. Определить, есть ли в данном двумерном массиве строка, состоящая только из элементов равных числу A .
19. Определить, есть ли в данном двумерном массиве столбец, состоящая только из элементов равных квадрату числа A .
20. Определить, есть ли в данном двумерном массиве строка, состоящая только из элементов, принадлежащих промежутку от A до B .
21. Определить, есть ли в данном двумерном массиве столбец, состоящий только из элементов, принадлежащих промежутку от A до B .
22. В каждой строке матрицы сменить знак максимального по модулю элемента на противоположный.
23. В каждом столбце матрицы сменить знак максимального по модулю элемента на противоположный.
24. Последний отрицательный элемент каждого столбца двумерного массива заменить нулём.
25. Положительные элементы двумерного массива умножить на первый элемент соответствующей строки, а отрицательные – на последний, то есть положительные элементы первой строки умножаем на первый элемент первой строки, а отрицательные – на последний элемент также первой строки, то же самое и с остальными строками.
26. Заменить все элементы двумерного массива строки с номером k и столбца с номером l на противоположные по знаку (элемент стоящий на пересечении, не изменять).
27. Составить программу заполнения и вывода на экран таблицы Пифагора (умножения).
28. Составить программу вывода на экран арифметического квадрата, в нём первый столбец и первая строки заполнены 1, а каждый из остальных элементов равен сумме своих соседей сверху и слева.

29. Заполнить массив A размером $n * m$ следующим образом, например, для $n=4$ и $m=7$:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

30. Заполнить массив A размером $n * m$ следующим образом, например, для $n=4$ и $m=7$:

1	0	2	0	3	0	4
0	5	0	6	0	7	0
8	0	9	0	10	0	11
0	12	0	13	0	14	0

31. Заполнить квадратный массив B размерностью $n * m$ следующим образом, например, для $n=6$:

1	12	13	24	25	36
2	11	14	23	26	35
3	10	15	22	27	34
4	9	16	21	28	33
5	8	17	20	29	32
6	7	18	19	30	31

32. Заполнить квадратный массив B размерностью $n * m$ следующим образом, например, для $n=6$:

1	3	4	10	11	21
2	5	9	12	20	22
6	8	13	19	23	30
7	14	18	24	29	31
15	17	25	28	32	35
16	27	27	33	34	36

33. Поменять местами первый максимальный и последний минимальный элементы.

34. В заданной строке поменять местами первый элемент и максимальный по

модулю.

35. В заданной строке переставить первый отрицательный и последний положительный элементы, если таких нет, то сообщить об этом.

36. Переставить вторую и предпоследнюю строки.

37. Поменять местами первую строку и строчку, в которой находится первый нулевой элемент.

38. Начиная с k -го столбца, сдвинуть их назад, а последние k поставить на место первых.

39. Начиная с k -го столбца, сдвинуть их вперед, а первые k поставить на место последних.

40. Дан двумерный массив. Начиная с первой строки, сдвинуть все строки на две вниз, а последние две перенести на место первых двух строк.

Список литературы

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня. - СПб: Питер, 2004г. - 461с.
2. Хохлов Д.Г. Основы технологии модульного программирования. Учебное пособие. - Казань. КГТУ (КАИ), кафедра АСОИУ, 2003г. - 63с.
3. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: Практикум. - СПб: Питер, 2002 г. - 240с.
4. Хохлов Д.Г. Структуры данных и комбинаторные алгоритмы. Учебное пособие. - Казань: КГТУ (КАИ), кафедра АСОИУ, 2003г.
5. Хохлов Д.Г., Захарова З.Х. Практикум по структурам данных и комбинаторным алгоритмам: Учебное пособие. - Казань: КГТУ (КАИ), кафедра АСОИУ, 2000г.
6. Бикмурзина А.Р. Лабораторный практикум по программированию. - Казань: КГТУ, 2000г.

Оглавление

Введение	6
1. Основные понятия программирования	7
2. Программирование линейных алгоритмов на языке Си	8
2.1. Способы записи алгоритма	8
2.2. Пример	9
2.3. Задачи	11
3. Операторы ветвления	13
3.1. Оператор if	13
3.2. Оператор switch	14
3.3. Примеры	15
3.4. Задачи	17
4. Операторы цикла	21
4.1. Цикл с параметром for	21
4.2. Цикл с предусловием while	22
4.3. Цикл с постусловием do while	22
4.4. Пример	23
4.5. Задачи	24
5. Одномерные массивы	27
5.1. Пример	27
5.2. Задачи	28
6. Двумерные массивы	31
6.1. Пример	31
6.2. Задачи	32
Список литературы	37
Оглавление	38

Ефимова Ю.В.

Практикум по программированию на языке высокого уровня

Учебное пособие

Ответственный за выпуск

Технический редактор
Компьютерная верстка –
ЛР № 020678 от 09.12.97

Подписано в печать

Формат 60×84 1/16. Бумага офсетная. Печать офсетная.
Печ.л. 3,0. Усл.печ.л 2,79. Усл.кр.-отт. 2,79. Уч.-изд.л. 2,63.
Тираж 100. Заказ Г 9/Д 83.

Издательство Казанского государственного технического	университета
Типография Издательства Казанского государственного	технического
университета	
420111 Казань, К. Маркса, 10	
