



Министерство образования и науки Украины
Севастопольский национальный технический университет

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ СИ++

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению контрольной работы № 2
по дисциплине
"ПРОГРАММИРОВАНИЕ "
для студентов заочной формы обучения
направления 6.050102
«Компьютерная инженерия»

Севастополь
2009



УДК 519.6

Программирование на языке СИ++. Методические указания к выполнению контрольной работы №2 по дисциплине «Программирование» для студентов заочной формы обучения направления 6.050102 «Компьютерная инженерия» / составители С.А.Бражников, Л.Н.Нежданова, С.Н.Фисун. - Севастополь: Изд-во СевНТУ, 2009.-24с.

Методические указания содержат требования к выполнению работы, варианты индивидуальных заданий, примеры составления программ и требования к оформлению контрольной работы.

Методические указания рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники, протокол №10 от 28 ноября 2008 г.

Рецензент:

Карапетьян В.А., [канд.техн.наук](#), доцент кафедры технической кибернетики

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Ответственный за выпуск: заведующий кафедрой кибернетики и вычислительной техники проф. Скатков А.В.

Отформатировано: Начать раздел:
на текущей странице

СОДЕРЖАНИЕ

1. Требования к выполнению работы	4
2. Задание 1 Разветвляющиеся вычислительные	5
3. Задание 2 Табулирование	6
4. Задание 3 Обработка одномерного массива	8
5. Задание 4 Подпрограммы	10
6. Задание 5 Обработка многомерных массивов	12
7. Задание 6 Обработка символьной информации	14
8. Задание 7 Текстовые файлы	16
9. Задание 8 Простые методы сортировки	18
10. Задание 9 Структуры	20
Библиографический список	23

1. ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ РАБОТЫ

Контрольные задания для студентов заочной формы обучения групп М-13 содержат 9 заданий по 25 вариантов в каждом. Все варианты одного задания имеют примерно одинаковую трудоемкость.

Вариант задания выбирается по двум последним цифрам номера зачетной книжки. Вариант = (остаток от деления двух последних цифр на 25)+1. Пусть номер 92087. Тогда $(87 \bmod 25) + 1 = 12 + 1 = 13$, т.е. номер вашего варианта 13.

Задания предназначены для изучения программирования на языке Си. Допускается использования любой интегрированной среды (Borland C++, C++, Turbo C++ и т.д.)

В течение семестра студенты выполняют 1 контрольную работу.

В приложениях к заданиям приведены примеры программ, поясняющих некоторые особенности языка C++. Использование этих примеров как прототипов **совершенно необязательно**. Например, вы можете использовать форматированный ввод-вывод, разработать более рациональный алгоритм и т.п.

Контрольные работы оформляются в отдельной тетради или на скрепленных жестко листах и содержат:

- **титульный лист** (с обязательным указанием номера зачетной книжки и варианта)

- **результаты выполнения девяти заданий**

Для каждого задания должны быть приведены:

- постановка задачи;
- алгоритм решения задачи (графическая схема алгоритма);
- описание входных и выходных данных (назначение, тип);
- -описания нестандартных функций- (назначение, входные и -выходные данные, способ передачи данных);
- -описание библиотечных функций (назначение, -имя заголовочного-файла). Если библиотечные функции используются в нескольких заданиях (например функции ввода, вывода-), достаточно представить описания таких функций один раз.
- текст программы
- результаты отладки программы

- Контрольные работы выполняются в -течение -семестра -и- привозятся на сессию.

Отформатировано: По ширине

Отформатировано: По ширине,
Отступ: Первая строка: 0,63 см

Отформатировано: По ширине

Отформатировано: По ширине,
Отступ: Слева: 0,63 см, Выступ: 0,62 см

Отформатировано: По ширине

2. ЗАДАНИЕ 1

Разветвляющиеся вычислительные процессы

Постановка задачи:

Составить программу решения выбранной по варианту задачи.

Исходные данные вводить с экрана. Результат решения выводить на экран.

Рекомендуется использовать потоковый ввод-вывод. Типы данных выбирать в соответствии с задачей.

1. Вычислить

$$Z = \begin{cases} \sin(x), & \text{если } x \leq a \\ \cos(x), & \text{если } a < x < b \\ \operatorname{Tg}(x), & \text{если } x \geq b \end{cases}$$

2. Прямая линия задана уравнением $Y=K \cdot X+B$. Определить, лежит ли точка с координатами X, Y ниже этой прямой.

3. Ввести M - номер месяца. Определить, входит ли этот месяц во 2-ой квартал

4. Определить, попадает ли точка с координатами X, Y в круг с радиусом R и с центром в начале координат.

5. Ввести G - год. Определить сколько дней в этом году, проверив предварительно является ли он високосным.

6. Ввести N - номер семестра. Определить, весенний это семестр или осенний.

7. Ввести целое число X . Определить, кратно ли оно 7 или 5.

8. Ввести X . Вычислить $Y=\ln(7X-1)$, если X входит в область допустимых значений (ОДЗ) этой функции. Если X вне ОДЗ, то вывести ответ “ X вне ОДЗ”.

9. Определить, расположена ли точка с координатами X, Y во II квадранте.

10. Ввести M - номер месяца. Определить, входит ли этот месяц в 1-ый квартал.

11. Прямая линия задана уравнением $Y=K \cdot X+B$. Определить, лежит ли точка с координатами X, Y выше этой прямой.

12. Ввести целое число X . Определить, кратно ли оно 3 или 7.

13. Ввести X . Вычислить $Y=\ln(0.5X-1)$, если X входит в ОДЗ этой функции. Если X вне ОДЗ, то вывести ответ “ X вне ОДЗ”.

14. Определить, расположена ли точка с координатами X, Y в IV квадранте.

15. Ввести M - номер месяца. Определить, входит ли этот месяц в 4-ый квартал

16. Ввести координаты двух точек $(X1, Y1)$ и $(X2, Y2)$. Определить, какая из них ближе к началу координат.

17. Вычислить

$$Y = \begin{cases} \sin(0.5x), & \text{если } x \leq a \\ \cos(2x), & \text{если } a < x < b \\ \operatorname{Tg}(x), & \text{если } x \geq b \end{cases}$$

18. Ввести M - номер месяца. Определить, к какому полугодию относится этот месяц.

19. Задать R - радиус круга и A - сторону квадрата. Определить площадь какой фигуры больше: круга с радиусом R или квадрата со стороной A .

20. Ввести G - год. Определить принадлежит ли он 17-му веку.

6

21. Задать R - радиус круга и A - сторону квадрата. Определить, верно ли, что площадь круга больше площади квадрата.
22. Ввести M - номер месяца. Определить, верно ли, что это месяц второго полугодия.
23. Ввести N - номер семестра. Верно ли, что это 2-й курс.
24. Ввести G - год. Определить к какому веку и какому тысячелетию относится этот год.
25. Ввести M - номер месяца. Определить, к какому кварталу относится этот месяц.

Приложение

// Определить, расположена ли точка с координатами X,Y в III квадранте

```
#include <iostream.h> // подключаем файл iostream.h
```

```
void main(void)
```

```
{ float y,x;
```

```
cout<<"\n"<<" введи y,x"<<endl;
```

```
cin>>y>>x; // ввод y, x
```

```
if ((x>0)&&(y<0)) cout<<" точка в III квадранте"<<"\n";
```

```
else cout<<"точка не в III квадранте "<<"\n";
```

```
}
```

3. ЗАДАНИЕ 2

Табулирование функции

Постановка задачи:

Вычислить таблицу значений функции $Y=F(X)$ для значений аргумента X, изменяющегося в интервале от $X_{нач}$ до $X_{кон}$ с шагом dX. [Смотри таблицу 3.1.](#)

Предусмотреть проверку ОДЗ функции. Если X вне ОДЗ, в таблицу выводить вместо значения Y – «нет решения».

Таблица [3.1](#)

Вариант	Вид функции $Y=F(X)$	$X_{нач}$	$X_{кон}$	dX
1	$0,7(1/x+\ln x)$	-16	34	4,5
2	$0.5x(2/x+4)$	0	15	1,5
3	$\sin^2(2x-0,7)+(1/x+2)$	-5	5	0,5
4	$0.7(1/x+2)-\cos(3x+0,7)$	-12	10	2
5	$x+(\operatorname{tg}(x+1))-2$	-20	10	3
6	$\sin(2x-0,7)(1/x+2)$	-4	4	0,8
7	$0.7(1/x+2)$	-10	10	2,5
8	$0,7(1/x+\ln x)$	-16	34	4,5
9	$1/x+(2x+4)$	0	15	1,5
10	$\sin(2x-0,7)+(1/x+2)$	-5	5	0,5

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 3.1

Отформатировано: Шрифт: 14 пт

11	$0.7(1/x+2) - \cos(3x+0,7)$	-12	10	2
12	$x^3 + (\operatorname{tg}(x+1) - 2)$	-20	10	3
13	$\operatorname{Sin}(2x-0,7)(1/x+2)$	-4	4	0,8
14	$x + \operatorname{Ln}(0.7x-1.5)$	-10	10	2,5
15	$0.7(1/x^2 + \operatorname{Ln}x)$	-16	34	4,5
16	$x + (2x^3 + 4)$	0	15	1,5
17	$\operatorname{Sin}(2x-0,7) + (1/x+2)$	-5	5	0,5
18	$0.7(1/x+2) - \cos(3x+0,7)$	-12	10	2
19	$\operatorname{tg}(x+1) - 2$	-20	10	3
20	$\operatorname{Sin}(2x-0,7)(1/x+2)$	-4	4	0,8
21	$0.7(1/x+2)$	-10	10	2,5
22	$(\operatorname{tg}(x+1) - 2)$	-20	10	3
23	$\operatorname{Sin}(2x-0,7)(1/x+2)$	-4	4	0,8
24	$\operatorname{Ln}(3x-2,5) + 7(\cos(x) - 2)$	-10	10	2,5
25	$\operatorname{Sin}(2x-0,7)(1/x+2)$	-4	4	0,8

Приложение

// Вывести на экран таблицу $F(X) = \sin(1/X)$ при $X_{\text{нач}}=0$ $X_{\text{кон}}=7,5$ $dX=0,5$

// исходные данные задать в программе

текст программы	комментарии
<code>#include <iostream.h></code>	
<code>#include <iomanip.h></code>	
<code>void main(void)</code>	
<code>{ double x0=0, xk=7.5, dx=0.5;</code>	инициализация x_0 , x_k , dx
<code>double x=x0, F;</code>	
<code>cout<<"\n"<<"табулирование функ-</code>	
<code>ции"<<endl;</code>	
<code>cout<<setw(6)<<"x"<<setw(14)<<"y"<<en</code>	вывод заголовка
<code>dl;</code>	таблицы
<code>while (x<=xk) {</code>	цикл с предуслови-
	ем: операторы в { }
	будут выполняться
	пока $x \leq x_k$
<code>if (x!=0)</code>	проверка ОДЗ
<code>{F=sin(1/x);</code>	вычисление F
<code>cout<<setprecision(3)<<x<<setw(14)<<F</code>	вывод на экран x (3
<code><<endl;</code>	знака после запятой), F
<code>}</code>	
<code>Else</code>	

<code>cout<<setprecision(3)<<x<<setw(14)<<" x вне ОДЗ"<<endl;</code>	вывод на экран x , " x вне ОДЗ"
<code>x+=dx;</code>	увеличение x на dx
<code>}}</code>	

4. ЗАДАНИЕ 3

Обработка одномерного массива

Постановка задачи:

Создать программу, позволяющую задать одномерный массив X , содержащий вещественные числа и выполнить обработку массива по варианту. Исходный массив и результаты **вывести на экран**.

Для отладки программы подготовить тестовые примеры с указанием размерности массива X и значений его элементов.

Таблица 4.1

Вариант	Содержание задания
1	Найти минимальное и максимальное число массива
2	Преобразовать массив, заменив все отрицательные числа на -1 , а положительные на $+1$. Нули оставить нулями.
3	Преобразовать массив, умножив все элементы массива на минимальный элемент этого массива
4	Преобразовать массив, умножив все элементы массива на максимальный элемент этого массива
5	Найти произведение всех положительных чисел массива и сумму отрицательных чисел
6	Преобразовать массив, переставив все числа в обратной последовательности
7	Сформировать новый массив, переписав в него сначала все положительные числа, затем все отрицательные, не изменяя их последовательности
8	Преобразовать массив, умножив все элементы массива на минимальный элемент этого массива
9	Найти сумму положительных чисел, расположенных до первого отрицательного числа
10	Проверить является ли массив упорядоченным по возрастанию
11	Сформировать новый массив, переписав в него сначала все отрицательные числа, затем все положительные, не изменяя их последовательности
12	Преобразовать массив, разделив все элементы массива на максимальный элемент этого массива

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 4.1

13	Преобразовать массив, разделив все элементы массива на минимальный элемент этого массива (если минимальный элемент массива равен 0, массив не изменять)
14	Найти сумму положительных чисел, расположенных до первого отрицательного числа
15	Проверить является ли массив упорядоченным по убыванию
16	Преобразовать массив, заменив все положительные числа массива на среднее арифметическое положительных чисел этого массива
17	Преобразовать массив, заменив все положительные числа массива на максимальный элемент этого массива
18	Преобразовать массив, заменив все отрицательные числа массива на среднее арифметическое положительных чисел этого массива
19	Преобразовать массив, разделив все элементы массива на среднее арифметическое положительных чисел этого массива
20	Найти сумму положительных чисел, расположенных до первого отрицательного числа
21	Преобразовать массив, заменив все отрицательные числа массива на минимальный элемент этого массива
22	Найти сумму положительных чисел, расположенных до первого отрицательного числа
23	Проверить является ли массив упорядоченным по возрастанию
24	Сформировать новый массив, переписав в него сначала все отрицательные числа, затем все положительные, не изменяя их последовательности
25	Преобразовать массив, разделив все элементы массива на максимальный элемент этого массива

Отформатировано: Шрифт: 14 пт

Отформатированная таблица

Приложение

// Заполнить массив случайными числами и вывести на экран.

#include <iostream.h>

#include <stdlib.h>

const int n=10;

void main(void) {

float a[n];

randomize;

for (int i=0; i<n; i++) a[i]=random(100)+1; // массив a заполняется случайными числами

cout<<"\n"<<"полученный массив"<<endl;

for (i=0; i<n; i++) cout<<"*(a+i)<<"\t"; // числа разделяются символом табуляции

cout<<endl;

}

5. ЗАДАНИЕ 4

Подпрограммы

Постановка задачи:

Создать программу, позволяющую задать одномерный массив X и выполнить его обработку по варианту. Исходный массив и результаты вывести на экран или во внешний файл.

Алгоритм обработки массива оформить как подпрограмму. Данные в подпрограмму передавать **только через параметры функции**. Саму подпрограмму поместить в другой файл. Оба файла должны размещаться в одной папке.

Т.о. проект будет состоять из двух файлов.

Для отладки программы подготовить тестовые примеры с указанием размерности массива X и значений его элементов. Тип элементов массива выбрать применительно к условию задания (так например, проверка кратности чисел применима только для целых чисел, а результат деления всегда будет вещественным и т.д.). Исходный массив X задать инициализацией, или генератором случайных чисел, или читать из внешнего файла.

Таблица [5.1](#)

Вариант	Содержание задания
1	Вычислить сумму всех чисел, кратных 7 или 5
2	Преобразовать массив, умножив все элементы массива на максимальный элемент этого массива
3	Найти среднее арифметическое всех положительных чисел массива и сумму отрицательных чисел
4	Преобразовать массив, переставив все числа в обратной последовательности
5	Сформировать новый массив, переписав в него сначала все положительные числа, затем все отрицательные, не изменяя их последовательности
6	Преобразовать массив, заменив все отрицательные элементы массива на минимальный элемент этого массива
7	Найти сумму положительных чисел, расположенных до первого отрицательного числа
8	Проверить является ли массив упорядоченным по возрастанию
9	Сформировать новый массив, переписав в него сначала все отрицательные числа, затем все положительные, не изменяя их последовательности
10	Преобразовать массив, увеличив все элементы массива на максимальный элемент этого массива
11	Преобразовать массив, уменьшив все элементы массива на минимальный элемент этого массива

Отформатировано: По левому краю

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 5.1

12	Найти сумму положительных чисел, расположенных до первого отрицательного числа
13	Проверить является ли массив упорядоченным по убыванию
14	Преобразовать массив, заменив все положительные числа массива на среднее арифметическое положительных чисел этого массива
15	Преобразовать массив, умножив все элементы массива на максимальный элемент этого массива
16	Найти произведение всех положительных чисел массива и сумму отрицательных чисел
17	Преобразовать массив, переставив все числа в обратной последовательности
18	Сформировать новый массив, переписав в него сначала все положительные числа, затем все отрицательные, не изменяя их последовательности
19	Преобразовать массив, разделив все элементы массива на минимальный элемент этого массива
20	Найти сумму положительных чисел, расположенных до первого отрицательного числа
21	Проверить является ли массив упорядоченным по возрастанию
22	Сформировать новый массив, переписав в него сначала все отрицательные числа, затем все положительные, не изменяя их последовательности
23	Преобразовать массив, разделив все элементы массива на максимальный элемент этого массива
24	Преобразовать массив, увеличив все элементы массива на минимальный элемент этого массива
25	Найти среднее арифметическое положительных чисел, расположенных до первого отрицательного числа

Отформатировано: Шрифт: 14 пт

Отформатированная таблица

Приложение

```
// Преобразовать массив, заменив все отрицательные числа массива на -1.
// Как функции оформлены: заполнение массива случайными числами, вывод массива,
// преобразование массива.
#include <iostream.h>
#include <stdlib.h>
#include "my_file.cpp"
void gener_mas( int *,int);      // прототип функции заполнения массива
const int n=15;
void main(void){
int *x=new int[n];
gener_mas(x,n);
cout<<"\n"<<" первоначальный массив"<<endl;
output_mas(x,n);
```

```

12
    change_mas(x,n);
cout<<"\n'<<" преобразованный массив"<<endl;
output_mas(x,n);
}
void gener_mas( int *b, int m){
b=new int[m];
for (int j=0;j<m;j++) *(b+j)=random(150)-75;    /* массив заполняется числами от -75
до +74 */
}
В файл my_file.cpp поместить описания функций output_mas и change_mas
void output_mas(int *b,int m){
for(int i=0;i<m;i++) cout<<b[i]<<"\t";
cout<<endl;
}
void change_mas(int *b,int m){
int pb,pbk;
pbk=b+m;                                     // вычисление адреса за последним элементом b
for(pb=b;pb<pbk;pb++) if (*pb<0) *pb=0;    // Замена отрицательных чисел массива в
на 0
}

```

6. ЗАДАНИЕ 5

Обработка многомерных массивов

Постановка задачи:

Создать программу, позволяющую задать двумерный массив, содержащий вещественные числа. Выполнить обработку элементов двумерного массива.

Исходный массив и результаты вывести на экран или во внешний файл. Двумерный массив выводить в виде таблицы.

Сам исходный массив задать инициализацией или генератором случайных чисел, или читать из внешнего файла.

Таблица [6.1](#)

Вариант	Содержание задания
1	Преобразовать массив, умножив элементы каждой строки на минимальный элемент этой строки
2	Сформировать массив, состоящий из максимальных чисел каждого столбца
3	Преобразовать массив: строку, в которой находился минимальный элемент массива заполнить нулями

Отформатировано: По левому краю

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 6.1

4	Преобразовать массив, элементы того столбца, в котором находился максимальный элемент, заменить на нули.
5	Преобразовать массив, увеличив элементы каждого столбца на последний элемент этого столбца.
6	Сформировать массив, состоящий из максимальных чисел каждой строки
7	Преобразовать массив, элементы строки, в которой находился максимальный элемент массива, заменить на нули
8	Преобразовать массив, элементы того столбца, в котором находился максимальный элемент, заменить на нули.
9	Преобразовать массив, уменьшив элементы каждой строки на максимальный элемент этой строки
10	Сформировать массив, состоящий из средних арифметических чисел каждого столбца
11	Преобразовать массив, умножив элементы каждой строки на минимальный элемент этой строки
12	Сформировать массив, состоящий из максимальных чисел каждого столбца
13	Сформировать массив, состоящий из максимальных чисел каждой строки
14	Преобразовать массив, элементы того столбца, в котором находился максимальный элемент, заменить на 1.
15	Преобразовать массив, увеличив элементы каждого столбца на последний элемент этого столбца.
16	Сформировать массив, состоящий из максимальных чисел каждой строки
17	Преобразовать массив, заменив все отрицательные числа на -1, а положительные на +1.
18	Преобразовать массив, элементы того столбца, в котором находился максимальный элемент, заменить на нули.
19	Преобразовать массив, уменьшив элементы каждой строки на максимальный элемент этой строки
20	Сформировать массив, состоящий из средних арифметических чисел каждого столбца
21	Сформировать массив, состоящий из средних арифметических чисел каждой строки
22	Преобразовать массив, умножив элементы каждой строки на минимальный элемент этой строки
23	Сформировать массив, состоящий из максимальных чисел каждого столбца

Отформатировано: Шрифт: 14 пт

Отформатированная таблица

Продолжение таблицы 6.1

24	Преобразовать массив, удалив строку, в которой находился минимальный элемент массива
25	Преобразовать массив, заменив все отрицательные числа на -1, а положительные на +1.

Отформатировано: Шрифт: 14 пт

Отформатированная таблица

Приложение

// Задать двумерный массив инициализацией. Найти максимальное число массива

#include <iostream.h>

#include <stdlib.h>

const n=2, m=3;

void main()

```
{ float mas[n][m]=((45,-6,8),
                    (0,458,-999));
```

int i,j;

cout<<"\n исходный массив \n";

for (i=0;i<n;i++){ // вывод на экран двумерного массива

for(j=0;j<m;j++) cout<<setw(8)<<mas[i][j];

cout<<"\n";

float maximum=mas[0][0];

for(i=0;i<n;i++)

for(j=0;j<m;j++)

if (mas[i][j]<maximum) maximum=mas[i][j];

cout<<"\n максимальное число ="<<max<<endl;

}

7. ЗАДАНИЕ 6**Обработка символьной информации****Постановка задачи:**

Задать символьную строку, состоящую из слов. Слова разделены пробелами.

Исходную строку символов ввести с экрана.

Выполнить обработку по варианту. Исходную строку и результат вывести на экран.

Подпрограмму обработки строки следует оформить в виде функции. При необходимости отдельные действия для обработки строки также следует оформить как функции.Таблица 7.1

Вариант	Содержимое задания
1	Преобразовать строку, заменив окончания слов "и" на "ы". Например, "наши сани" на "нашы саны"
2	Преобразовать строку, заменив все слова "враг" на слово "друг"
3	Преобразовать строку, удалив из нее все трехбуквенные слова

Отформатировано: По левому краю

Отформатированная таблица

Продолжение таблицы 7.1

4	Преобразовать строку, заменив все слова “государство” на слово “страна”
5	Сжать строку, заменив любую группу пробелов одним пробелом.
6	Преобразовать строку, заменив все слова ” государство” на слово “страна”
7	Преобразовать строку, заключив все трехбуквенные слова в круглые скобки
8	Преобразовать строку, заменив все слова “государство” на слово “страна”
9	Сжать строку, заменив любую группу пробелов одним пробелом.
10	Преобразовать строку, заменив все слова ” сын” на слово “дочь”
11	Преобразовать строку, удалив из нее все четырехбуквенные слова
12	Преобразовать строку, заменив все слова “государство” на слово “страна”
13	Сжать строку, заменив любую группу пробелов одним пробелом.
14	Преобразовать строку, заменив все слова ” государство” на слово “страна”
15	Преобразовать строку, заключив все сочетания “авс” на ”ghn”
16	Преобразовать строку, заменив все слова “государство” на слово “Украина”
17	Сжать строку, заменив любую группу одинаковых символов одним символом.
18	Преобразовать строку, удалив из нее все цифры
19	Определить количество слов, начинающихся на ‘А’ и заканчивающихся на “ и”
20	Преобразовать строку, удалив слова состоящие только из цифр
21	Определить количество двухбуквенных слов
22	Преобразовать строку, удалив слова состоящие только из гласных букв
23	Преобразовать строку, заменив слова состоящие только из гласных букв на пробелы
24	Преобразовать строку, заменив все слова ” государство” на слово “страна”
25	Преобразовать строку, заменив все сочетания “авс” на ”1234”

Приложение

// Ввести символьную строку, состоящую из слов. Слова разделены группой пробелов.

// Определить количество четырехбуквенных слов

// Количество таких слов определяется в функции `count_word`

#include <iostream.h>

#include <string.h>

16

```
int count_word(char *stroka){
int k=0; char *word , Razd[]=" ";
word=strtok(stroka,Razd);           // выделение 1-го слова
while(word){
if(strlen(word)==4) k++;
word=strtok(NULL,Razd);           // выделение очередного слова
return k;}
void main(){
char STROKA[80];
cin.getline(STROKA,80);
cout<<" в этой строке таких слов "<<count_word(STROKA)<<"\n";
}
```

8. ЗАДАНИЕ 7

Текстовые файлы

Постановка задачи:

Разработать программу, которая выполняет следующие действия: создает **входной поток** для построчного чтения из внешнего файла и **выходной поток** для вывода в выходной файл. Текст из входного потока извлекается построчно, выполняется обработка прочитанной строки согласно варианту задания, результат записывается в выходной файл.

Каждая прочитанная строка состоит из слов. Слова разделены пробелом.

Подпрограмму обработки строки оформить в виде функции. При необходимости отдельные действия для обработки строки также следует оформить как функции.

Таблица 8.1

Вариант	Содержание задания
1	Подсчитать сколько слов в каждой строке состоит только из цифр
2	Все четырехбуквенные слова заключить в круглые скобки
3	Сжать строку, заменив любую группу пробелов одним пробелом.
4	Преобразовать строку, заменив все слова "сын" на слово "дочь"
5	Преобразовать строку, удалив из нее все четырехбуквенные слова
6	Преобразовать строку, заменив все слова "государство" на слово "страна"
7	В каждой строке подсчитать количество слов, в которых первый и последний символы одинаковы..
8	Преобразовать строку, заменив все слова "государство" на слово "страна"

Отформатировано: По левому краю

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 8.1

9	Преобразовать строку, заключив все сочетания “авс” в квадратные скобки.
10	Преобразовать строку, заменив все слова “государство” на слово “Украина”
11	Сжать строку, заменив любое слово из одинаковых символов одним символом.
12	Преобразовать строку, удалив из нее все цифры
13	Определить количество слов в каждой строке, начинающихся на ‘А’ и заканчивающихся на “ и”
14	Преобразовать строку, удалив слова состоящие только из цифр
15	Определить количество двухбуквенных слов в каждой строке
16	Преобразовать строку, удалив слова состоящие только из гласных букв
17	Преобразовать строку, перед каждым словом “страна” вставить слово “наша”
18	Преобразовать строку, заключив все трехбуквенные слова в круглые скобки
19	Преобразовать строку, заменив все слова “государство” на слово “страна”
20	Сжать строку, заменив любую группу пробелов одним пробелом.
21	Преобразовать строку, заменив все слова ”сын” на слово “дочь”
22	Преобразовать строку, удалив из нее все четырехбуквенные слова
23	Преобразовать строку, заменив все слова “государство” на слово “страна”
24	Преобразовать строку, удалив слова состоящие только из цифр
25	Определить количество двухбуквенных слов в каждой строке

Приложение

// Задан текстовый файл. Прочитать построчно, и преобразовать каждую строку, заменив все слова “да” на “нет”.

// Результат вывести во внешний файл.

```
#include <fstream.h>
```

```
#include <string.h>
```

```
char *change_str(char *);
```

```
void main(){
```

```
ifstream fin(“isx.txt”,ios::in);
```

```
ofstream fout(“rez.txt”,ios::out);
```

```
char *str;
```

```
while(!fin.eof()){
```

```
fin.getline(str,100);
```

```
fout<<change_str(str)<<endl;
```

```
}
```

```

18      fin.close();
      fout.close();
    }

    char *change_str(char *st){
    char *slovo, *new_st=" ";
    slovo=strtok(st, " ");
    while(slovo){
    if(strcmp(slovo,"da")) strcat(new_st, "nem "; else{ strcat(new_st,slovo); strcat(new_st, " ");}
    return new_st;
    }

```

9. ЗАДАНИЕ 8

Простые методы сортировки

Постановка задачи:

Создать программу, выполняющую сортировку массива заданным методом. Исходный массив и отсортированный вывести во внешний файл.

Алгоритм сортировки оформить в виде функции. Данные в подпрограмму передавать *только через параметры функции.*

Для отладки программы значения размерности массива и сами элементы массива придумать самим. Тип элементов массива выбрать согласно варианту.

Исходный массив читать из внешнего файла или задать генератором случайных чисел.

Таблица 9.1

Вариант	Метод сортировки	Тип элементов массива
1	Пузырьковая сортировка по возрастанию (прямое направление)–	целые
2	Пузырьковая сортировка по убыванию (обратное направление)–	целые без знака
3	Сортировка по возрастанию методом прямого выбора	целые
4	Сортировка по убыванию методом прямой вставки (включения)——	целые
5	Пузырьковая сортировка по убыванию (прямое направление)——	целые без знака
6	Пузырьковая сортировка по возрастанию (обратное направление)	символьные
7	Сортировка по убыванию методом прямого выбора——	вещественные
8	Сортировка по возрастанию методом прямой вставки (включения)	целые

Отформатировано: По левому краю

Отформатировано: По центру

Отформатированная таблица

Продолжение таблицы 9.1

9	Пузырьковая сортировка по возрастанию (обратное направление) —	вещественные
10	Пузырьковая сортировка по убыванию (прямое направление) —	вещественные
11	Сортировка по возрастанию методом прямого выбора	целые без знака
12	Сортировка по убыванию методом прямой вставки (включения) —	вещественные
13	Пузырьковая сортировка по убыванию (прямое направление) —	символьные
14	Пузырьковая сортировка по возрастанию (обратное направление)	вещественные с двойной точностью
15	Сортировка по убыванию методом прямого выбора	вещественные
16	Сортировка по возрастанию методом прямой вставки (включения)	вещественные
17	Пузырьковая сортировка по возрастанию (прямое направление)	символьные
18	Пузырьковая сортировка по убыванию (обратное направление)	Символьные без знака
19	Сортировка по возрастанию методом прямого выбора	символьные
20	Сортировка по убыванию методом прямой вставки со сдвигом —	символьные
21	Пузырьковая сортировка по убыванию (прямое направление) —	Символьные без знака
22	Пузырьковая сортировка по возрастанию (обратное направление)	символьные
23	Сортировка по убыванию методом прямого выбора —	целые без знака
24	Сортировка по возрастанию методом прямой вставки со сдвигом — е	символьные
25	Пузырьковая сортировка по возрастанию количества единиц в коде (прямое направление)	целые без знака

Простые методы сортировки:

Простые методы сортировки подробно описаны в [4].

Пусть массив a содержит n элементов и должен быть упорядочен по возрастанию.

Сортировка прямым выбором: состоит из $(n-1)$ шагов. Общий, i -й, шаг таков: выбирается наименьший среди элементов $a[i], a[i+1], \dots, a[n-1]$ и меняется местами с элементом $a[i]$. Общий шаг повторяется для $i=0, 1, \dots, n-2$.

Сортировка методом прямой вставки: Пусть элементы $a[0], a[1], \dots, a[i-1]$ уже рас-

положены правильно, а оставшиеся элементы еще не рассмотрены. Элемент $a[i]$ реносится в “карман”. Будем перемещаться от элемента $a[i-1]$ к началу массива. Все элементы, большие, чем $a[i]$ сдвигаются последовательно на одну позицию вправо, а правее первого из неподвижных элементов ставится $a[i]$. Теперь количество правильно расположенных элементов увеличилось на 1. Процесс повторяется для $i=1,2,\dots,n-1$.

Пузырьковая сортировка: состоит из $(n-1)$ шагов. На каждом шаге сравниваются последовательно 2 соседних элемента (пара) и при неправильном расположении меняются местами. Если на первом шаге рассматривались элементы $a[0], a[1], \dots, a[n-1]$ (прямое включение), то наибольший элемент попал на место $a[n-1]$. Значит, на следующем шаге надо рассматривать элементы $a[0], a[1], \dots, a[i-2]$ и т.д.

Если во внутреннем цикле двигаться от конца массива к началу (обратное включение), то нужный элемент перемещается к началу массива.

Приложение

// Написать функцию определения количества единиц в коде

```
int count_ed(unsigned int a){
int k=0;
while ( a!=0) { a&=(a-1); k++;}
return k;}
```

10. ЗАДАНИЕ 9

Структуры

Постановка задачи:

Задана предметная область для базы данных и предложена структура для описания отдельных записей этой базы. Как правило, структура имеет 3 поля (элементов) двух или трех типов.

Например: структура «государство»

Элементы (поля) структуры:

Название страны

Площадь территории

Столица

1 Создать одномерный массив структур. Не менее 5 структур. Значения полей вводить с клавиатуры. При вводе структур можно реализовать один из механизмов:

- ввод заранее заданного количества структур;
- ввод до появления структуры с заданным признаком;
- диалог с пользователем о необходимости продолжать ввод

2 Написать функцию поиска (и вывода во внешний файл) структуры по заданному признаку. Если структура с заданным значением поля отсутствует, вывести сообщение об этом. Например, найти структуру по размеру территории. Для каждой найденной в массиве структуры с такой площадью, вывести название этой страны, ее столицу, площадь территории. Если страна с такой площадью не найдена ни одна, можно вывести «страны с такой площадью нет».

3 Массив структур и результаты поиска вывести во внешний файл.

Варианты заданий:

Таблица 10.1

Вариант	Поля структуры	Поле, по значению которого искать структуру
1	Марка автомобиля, серийный номер, год выпуска	Серийный номер
2	Марка автомобиля, серийный номер, год выпуска	Год выпуска
3	Марка автомобиля, серийный номер, год выпуска	Марка автомобиля
4	Фамилия, год рождения, номер медицинской карты	Фамилия
5	Фамилия, год рождения, номер медицинской карты	Год рождения
6	Фамилия, год рождения, номер медицинской карты	Номер медицинской карты
7	Фамилия, номер телефона, номер кредитной карточки	Фамилия
8	Фамилия, номер телефона, номер кредитной карточки	Номер кредитной карты
9	Фамилия, номер телефона, номер кредитной карточки	Номер телефона
10	Город, количество жителей, количество военнообязанных	Количество военнообязанных
11	Город, количество жителей, количество избирателей	Количество жителей
12	Город, количество жителей, количество избирателей	Количество избирателей
13	Город, количество высших учебных заведений, количество жителей	Количество ВУЗов
14	Город, количество высших учебных заведений, количество жителей	Город
15	Город, количество высших учебных заведений, количество жителей	Количество жителей
16	Марка автомобиля, год выпуска, цена	Цена
17	Марка автомобиля, год выпуска, цена	Год выпуска
18	Марка автомобиля, год выпуска, цена	Марка автомобил
19	Фамилия, номер медицинской карты, возраст	Возраст
20	Название страны, государственный язык, количество жителей	Государственный язык

Отформатировано: По левому краю

Продолжение таблицы 10.1

21	Название страны, государственный язык, количество жителей	Название страны
22	Название фильма, страна, год выпуска	Страна
23	Название фильма, страна, год выпуска	Год выпуска
24	Компьютер, быстродействие, объем оперативной памяти	Объем оперативной памяти
25	Компьютер, год выпуска, объем оперативной памяти, цена	цена

Приложение

Создать массив структур (3 структуры). Структура имеет 2 поля: фамилия, возраст. Найти сведения по заданной фамилии. Массив структур заполняется с клавиатуры. Массив и результаты поиска выводятся во внешний файл. Приведен пример **простейшей** программы.

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
const n=3;
/* Определение структурного типа */
typedef struct {
char fam[9];
int year;} my_st;
void main(){
int flag=1;
char rr[9];
my_st *massiv;
massiv=new my_st[n];
ofstream fout("rez_struct",ios::out);
/* Создание массива структур, ввод с клавиатуры */
cout<<"\n vvesti <<n<<" structur \n";
for(int i=0;i<n;i++){
cout<<"\n input fam"; cin>>massiv[i].fam;
cout<<"\n input ye"; cin>>massiv[i].year;}
/* Вывод массива структур во внешний файл */
for(i=0;i<n;i++) fout<<massiv[i].fam<<"\t"<<massiv[i].year<<endl;
cout<<"Vvesti Famly for find ";cin>>rr;
fout<<"\n Resultat poiska\n";
/* Поиск структуры с заданной фамилией */
for(i=0;i<n;i++)
```

```

if(!strcmp(massiv[i].fam,rr))
{flag=0; fout<<massiv[i].fam<<"\t"<<massiv[i].year<<endl;}
if(flag)fout<<"\n no that family \n";
}

```

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 19.701-90. ЕСПД “Системы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения” Дата введения: 01.01.1992.-М.: Изд-во стандартов, 1991. - 26с.
2. Подбельский В.В. Программирование на языке Си: учеб.пособие / В.В. Подбельский, С.С. Фомин - М.: Финансы и статистика, 2002.-600с.
3. Уэйт М. Руководство для начинающих: [пер. с англ.]/ М. Уэйт, С. Прата, Д. Мартин - М.: Мир, 1988.-502с.
4. Хомоненко В.Е. С++: учебное пособие / В.Е. Хомоненко - М.: Финансы и статистика, 2004.- 540с.

Заказ № _____ от « ____ » _____ 200 ____ Тираж _____ экз.
Изд-во СевНТУ