

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з дисципліни
«Основи програмування»

для студентів
кафедри автоматики та управління в технічних системах

Укладач:
В.А. Жеребко, ст. викладач
кафедри АУТС, факультету ІОТ

Київ
НТУУ «КПІ»
2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ЛАБОРАТОРНАЯ РАБОТА № 1	4
Линейный вычислительный процесс	4
ЛАБОРАТОРНАЯ РАБОТА № 2	10
Операторы языка C++	10
ЛАБОРАТОРНАЯ РАБОТА № 3	17
Изучение функций в языке C++	17
ЛАБОРАТОРНАЯ РАБОТА № 4	24
Ввод, вывод. Файлы данных	24
ЛАБОРАТОРНАЯ РАБОТА № 5	29
Работа с массивами	29
ЛАБОРАТОРНАЯ РАБОТА № 6	34
Работа со строками	34
ЛАБОРАТОРНАЯ РАБОТА № 7	39
Указатели и ссылки	39
ЛАБОРАТОРНАЯ РАБОТА № 8	41
Работа с памятью	41

ВВЕДЕНИЕ

Программа подготовки бакалавров, специалистов и магистров по специальности «Компьютеризованные системы управления и автоматики» предусматривает изучение основ программирования и вычислительной техники.

В данных методических указаниях приведены задания к лабораторным работам по курсу «Основы программирования». В качестве языка программирования был выбран C++. Для решения задач, приведенных в данных методических указаниях, используются версии 3.1 и 4.5 компилятора C++ фирмы Borland International. Большинство заданий может быть выполнено с применением компиляторов других фирм. Данный язык является в настоящее время одним из самых распространенных и предоставляет широкие возможности для решения всех задач современного программирования.

Цель лабораторных работ:

- овладеть языком программирования C++;
- изучить и освоить основные приемы программирования;
- получить навыки самостоятельного решения задач.

Изучение языка программирования C++ предусматривается на протяжении двух семестров (второй семестр I курса и первый семестр II курса). Лабораторные работы предусматривают изучение основ программирования на C++.

Описание каждой лабораторной работы содержит краткие теоретические сведения, необходимые для решения задач и варианты заданий. Изложенные теоретические сведения к лабораторным работам не претендуют на полноту и для выполнения некоторых задач необходимо использование дополнительной литературы, список которой приведен в приложении.

ЛАБОРАТОРНАЯ РАБОТА № 1

Линейный вычислительный процесс

Краткие теоретические сведения

1.1 Пример программы

Пример программы, написанной на C++, выполняет подсчет пробелов во введенной пользователем строке.

```
// Подсчитывает пробелы в строке
#include <stdio.h> // данные заголовочные файлы имеются в
#include <string.h> // пакете Borland C++
                    // и содержит необходимые для
                    // компиляции программы объявления
const int TAB_SIZE=8;
int main ()
{
    char words [80];
    // получение строки от пользователя
    gets(words) ;
    int spaces = 0; // сначала количество пробелов = 0
    // подсчет пробелов
    for (int i = 0; i < strlen(words); i++)
    {
        switch (words [i])
        {
            case ' ':
                spaces++;
                break;
            case '\t':
                spaces += TAB_SIZE;
                break;
            default:
                break;
        }
    }
    // вывод результата
    printf("\n Число пробелов равно %d\n" , spaces);
    // функция main возвращает значение 0
    return 0;
}
```

Данная программа содержит следующие конструкции языка C++:

- Комментарии;
- Включение файлов;
- Функция main();
- Переменные;
- Операторы;

- Вызов функций;

1.2 Операторы

Для вычисления значений выражений используются операторы, операнды и вызовы функций:

- Арифметические операторы: $*$, $/$, $+$, $-$, $\%$ (деление по модулю);
- Операторы отношений: $<$, $>$, $<=$, $>=$, $==$ (равно), $!=$ (не равно);
- Логические операторы: $\&\&$ (И), $\|\|$ (ИЛИ), $!$ (НЕ);
- Операторы инкремента/декремента: $++i$ / $i++$; $--i$ / $i--$;
- Поразрядные операторы: $\&$ (И), $\|$ (ИЛИ), $^$ (Исключающее ИЛИ), $<<$ (сдвиг влево),
 $>>$ (сдвиг вправо), \sim (дополнение до 1).

1.3 Функции, используемые при вычислении выражений:

Функция в C++	Математический эквивалент
<code>int abs(int x)</code>	$ x $
<code>double fabs(double x)</code>	$ x $
<code>double acos(double x)</code>	$\arccos(x)$
<code>double asin(double x)</code>	$\arcsin(x)$
<code>double atan(double x)</code>	$\arctg(x)$
<code>double atan2(double y, double x)</code>	$\arctg(y/x)$
<code>double sin(double x)</code>	$\sin(x)$
<code>double cos(double x)</code>	$\cos(x)$
<code>double tan(double x)</code>	$\operatorname{tg}(x)$
<code>double cosh(double x)</code>	$\operatorname{ch}(x)$
<code>double sinh(double x)</code>	$\operatorname{sh}(x)$
<code>double tanh(double x)</code>	$\operatorname{th}(x)$
<code>double exp(double x)</code>	e^x
<code>double pow(double x, double y)</code>	x^y
<code>double log(double x)</code>	$\ln(x)$
<code>double log10(double x)</code>	$\lg(x)$
<code>double sqrt(double x)</code>	\sqrt{x}

Для использования математических функций необходимо включать файл **math.h**.

1.4. Типы данных

Типы данных, используемые в C++, диапазоны их значений и операторы вывода приведены в следующем примере:

```
#include <stdio.h>

int main ()
{
    // Типы данных
    // Signed char (1 byte, 8 bit, -128 ... +127)
```

```
signed char a=-34;
char c=28, cl='M';
printf ("Signed char a= %d c= %d \n", a, c);
printf ("Signed char (char) cl=%c \n", cl);
// Unsigned char (1 byte, 8 bit, 0 ... 255)
unsigned char b=232;
printf ("Unsigned char (value) b= %d \n", b);
printf ("Unsigned char (oktal) b= %o \n", b);
printf ("Unsigned char (hexadecimal) b= %X \n", b);
// Signed short (2 byte, 16 bit, -32768 ... 32767)
short d=-1234;
printf ("Signed short d= %hd \n", d);
// Unsigned short (2 byte, 16 bit, 0 ... 65535)
unsigned short e=42123;
printf ("Unsigned short e= %u \n", e);
// Signed int (2 byte, 16 bit, -32768 ... 32767)
int f=-4321;
printf ("Signed int f= %d \n", f);
// Unsigned int (2 byte, 16 bit, 0 ... 65535)
unsigned int g=34343;
printf ("Unsigned int g= %u \n", g);
// Signed long (4 byte, 32 bit, -2147483648 ... 2147483647)
long h=-67464432;
printf ("Signed long h= %ld \n", h);
// Unsigned long (4 byte, 32 bit, 0 ... 4294967295)
Unsigned long i=410000000;
printf ("Unsigned long i= %ld \n",I);
// Float (4 byte, 32 bit, 3.4E-38 ... 3.4E+38)
float j=1.12345678;
printf ("Float j = %f \n",j);
// Double (8 byte, 64 bit, 1.7E-308 double ... 1.7E+308)
double k=1.12345678901234567;
printf ("Double k= %f \n", k);
printf ("Double k= %20.16f \n",k);
printf ("Double k= %e \n", k)
printf ("Double k= %20.16e \n",k);
printf ("Double k= %E \n", k);
printf ("Double k= %g \n", k);
printf ("Double k= %G \n", k);
// Long double(10 byte, 80 bit, 3.4E-4932 ... 1.1E+4932)
long double l=1.123456789012345678;
printf ("Long double l= %21.18Lf \n",l);
return 0;
}
```

После запуска программы на экран выводятся следующие результаты:

```
Signed char a= -34 c= 28
Signed char (char) cl=M
Unsigned char (value) b= 232
Unsigned char (oktal) b= 350
```

```

Unsigned char (hexadecimal) b= E8
Signed short d= -1234
Unsigned short e= 42123
Signed int f= -4321
Unsigned int g= 34343
Signed long h= -67464432
Unsigned long i= 410000000
Float j= 1.123457
Double k= 1.123457
Double k= 1.123456789B123457
Double k= 1.123457e+00
Double k= 1.1234567890123457e+00
Double k= 1.123457E+00
Double k= 1.12346
Double k= 1.12346
Long double l= 1.123456789012345690
    
```

Задание на лабораторную работу № 1

Ввести, скомпилировать и выполнить программу подсчета количества пробелов в строке;

Выполнить свой вариант задания из таблицы 1 (для тригонометрических функций аргументы заданы в радианах);

Таблица 1 – Варианты заданий для лабораторной работы № 1

Вариант	Выражение	Значение параметров			
		a	b	c	d
1	$y = \sqrt{\left \sin(a) - \frac{4 \cdot \ln(b)}{c^d} \right }$	-1.49	23.4	1.23	2.542
2	$y = \frac{e^a + 3 \cdot \lg(c)}{\sqrt{b^c}} \cdot \operatorname{arctg}(d) $	2.34	0.756	2.23	-1.653
3	$y = (2 \cdot \sin(a) + \cos[b \cdot \sqrt{c}])^d$	1.234	-3.12	5.45	2.0
4	$y = \frac{\sqrt[3]{\operatorname{tg}(a)}}{1 + \frac{\operatorname{sh}(b)}{\ln d+c }}$	3.56	1.02	3	2.43
5	$y = 2 \cdot \cos(a^b) + \left \arccos\left(-\sqrt{\frac{c}{d}}\right) \right $	2.54	1.23	-2.14	-0.23
6	$y = 3 \cdot \left(\lg\left \frac{b}{a}\right + \sqrt{\sin(c) + e^d} \right)$	-1.23	-0.34	0.707	2.312
7	$y = \left(4 \cdot \operatorname{ch}\left(\sqrt{\frac{a}{b}}\right) + 3 \cdot \arccos(d) \right)^c$	-3.45	2.34	1.45	0.83

8	$y = \frac{5 \cdot a}{\sin(a)} + \sqrt{\frac{th(b \cdot c)}{\ln(d)}}$	0.345	-2.25	2.65	3.99
9	$y = \frac{\sin a + \cos(\sqrt{b})}{2 \cdot tg(c) + e^d}$	-2.98	5.55	0.045	0.129
10	$y = \frac{a^b}{sh(b)} + 4 \cdot \frac{\ln(c)}{\sqrt[4]{d}}$	1.27	10.99	2.73	25.32
11	$y = 9 \cdot \frac{\arcsin(a)}{\cos \sqrt{ b }} + 2.43^d + \ln(c)$	0.478	-1.26	2.68	18.24
12	$y = 6 \cdot (\sin 2a)^{\lg(b)} + \sqrt{c \cdot ch(-d)}$	1.478	9.26	0.68	2.24
13	$y = \frac{\ln 2a }{3 \cdot tg(b)} - \sqrt{\frac{c}{\sqrt{ d }}}$	-2.86	1.62	10.874	-2.91
14	$y = 2 \cdot \sqrt{\frac{\sin(a)}{ tg(b-a) } + \frac{\ln(c)}{d}}$	1.54	0.49	24.1	0.87
15	$y = 2 \cdot \sqrt{\frac{tg(a)}{ \arccos(b) }} - 3 \cdot \sqrt[3]{e^{c-a} \cdot sh(d)}$	1.25	3.09	4.25	0.56
16	$y = 2 \cdot \frac{\sin(a)}{\arccos(-2b)} - \sqrt{\ln(c \cdot 2d)}$	0.58	0.34	1.25	-1.89
17	$y = 2 \cdot \sqrt{tg a+c } + \frac{\ln(b)}{c^d}$	-1.49	23.4	1.23	2.254
18	$y = \frac{e^c + 2 \cdot \lg(a)}{\sqrt{c^b}} \cdot \arcsin(d) $	2.34	0.756	2.23	-1.653
19	$y = \frac{\sqrt[5]{tg(a)}}{1 + \frac{sh(b)}{\ln d+c }}$	1.234	-3.12	5.45	2.0
20	$y = 2 \cdot \ln(b^a) + \left \arcsin\left(-\sqrt{\frac{d}{c}}\right) \right $	3.56	1.02	3	2.43
21	$y = 3 \cdot \left(\ln\left \frac{a}{b}\right + \sqrt{\cos(c) + e^d} \right)$	-2.54	1.23	-2.14	-0.23
22	$y = \left(4 \cdot sh\left(\sqrt{\left \frac{a}{b}\right }\right) + 3 \cdot \arcsin(c) \right)^d$	1.23	-0.34	0.707	2.312
23	$y = \frac{5 \cdot c}{\cos(a)} + \sqrt{\frac{sh(b \cdot c)}{tg(d)}}$	-3.45	-2.34	1.45	0.83
24	$y = \frac{\cos(b) + \sin(\sqrt{a})}{2 \cdot \lg(c) + e^d}$	0.345	-2.25	2.65	3.99
25	$y = \frac{a^b}{ch(b)} + 4 \cdot \frac{\lg(c)}{\sqrt[5]{d}}$	2.98	5.55	0.045	0.129

26	$y = \frac{\sqrt[3]{tg(a)}}{2 - \frac{sh(b)}{\ln d+c }}$	1.27	10.99	4	-25.32
27	$y = \sin(a^{-b}) + 3 \cdot \left \arccos\left(-\sqrt{\frac{c}{d}}\right) \right $	0.478	-1.26	2.68	18.24
28	$y = 2 \cdot \left(\ln\left \frac{b}{a}\right + \sqrt{sh(c) + e^d} \right)$	1.478	9.26	0.68	2.24
29	$y = \left(2 \cdot \cos\left(\sqrt{\frac{a}{b}}\right) + 4 \cdot arcsh(d) \right)^c$	-2.86	1.62	10.874	2.91
30	$y = \frac{3 \cdot a}{\cos(a)} + \sqrt{\frac{th(b \cdot c)}{\ln(d)}}$	0.58	-0.34	1.25	1.89

ЛАБОРАТОРНАЯ РАБОТА № 2

Операторы языка C++

Краткие теоретические сведения

2.1 Оператор **if**

```
        if (выражение)
            оператор;
или
        if (выражение)
        {
            оператор;
            оператор2;
        }
```

Примеры:

```
if (a==b) printf ( a ) ;
if ( (s<=5) && (t==0) || (c!=d) )
    { c=12 ;
      d=18 ; }
```

2.2 Оператор **else**

```
        if(выражение)
            оператор1;
        else
            оператор2;
или
        if(выражение)
            {оператор1;
             оператор2;}
        else
            {оператор3;
             оператор4;}
```

Пример:

```
if (a<=b)

else
{
    a--;
    b++;
}
```

2.3 Оператор **switch** (аналог **case** в Паскале)

```
switch (выражение)
```

```
{
case значение1:
    оператор1;break;//выражение == значение1
case значение2:
    оператор2;break;//выражение == значение2
default:
    оператор3;
}
```

Пример:

```
switch (c)
{
case 1:
    a=5;break;
case 2:
    b=6;break;
default:
    a=b;
}
```

2.4 Оператор **while**

```
while (выражение)
    оператор;
или
while (выражение)
{
    оператор1;
    оператор2;
}
```

Пример:

```
while (int a<=10)
{
printf("%d\n",a);
a++;
}
```

2.5 Оператор **do-while** (аналог repeat-until в Паскале)

```
do
оператор;
while (выражение); //оператор выполняется, пока
                    //выражение истинно
или
do
{оператор1;
 оператор2;}
while (выражение);
```

Пример:

```
do
{a++;
printf ("%d\n", a); }
while (a<10);
```

2.6 Оператор **for**

```
for (выражение инициализации; выражение условия;
     выражение изменения)
оператор;
```

или

```
for (выражение_инициализации; выражение_условия;
     выражение изменения)
{оператор1;
 оператор2; }
```

Пример:

```
for (c=32; c<128; c++)
printf(" %c\n", c);
```

2.7 Оператор **break**

Используется для прерывания циклов **while**, **do-while**, **for**.

Переход осуществляется на первый оператор, расположенный за циклом.

Пример:

```
for (int
i=0; i<10; i++)
{
printf
("i=%d\n", i);
if (i>5) break;
}
```

Цикл выполнится до i=6 и будет прерван.

2.8 Оператор **continue**

Оператор **continue** прерывает текущую итерацию цикла и начинает следующую.

Пример:

```
for (int i=0; i<10; i++)
{
    if (i=5) continue; printf ("i=%d\n", i);
}
```

На экран будут выведены числа 0-9, за исключением 5.

2.9 Оператор **goto**

Оператор **goto** осуществляет переход на метку.

Пример:

```
int i=0;
```

```
m:
printf ("i=%d\n", i);
if(i<10) goto m;
```

В отличие от Паскаля метки определять не нужно. Использовать оператор **goto** не рекомендуется.

2.10 Ввод чисел

Для ввода чисел можно использовать следующие возможности:

а) Ввод строки с помощью функции **gets** и преобразование в число с помощью функции **atof**:

```
char a[30];
double b;
printf("Введите b:");
gets(a); b=atof(a);
```

б) С помощью функции **scanf**:

```
double s, t;
scanf ("%f %f", &s, &t);
```

в) С помощью потокового ввода (**cin**, **cout**).

Задание на лабораторную работу № 2

Ввести исходные данные, произвести их контроль (область допустимых значений аргументов). Для вычислений использовать все варианты циклов (while, do – while, for);

Выполнить свой вариант задания из таблицы 2;

Таблица 2 – Варианты заданий для лабораторной работы № 2

Вариант	Задание
1	Вычислить сумму ряда: $\sum_{i=1}^k \sqrt{m \cdot \frac{1}{i}} \cdot \sin(m \cdot i)$, $k \leq 30$
2	Вычислить сумму ряда: $\sum_{i=1}^k \ln(-t \cdot i) \cdot \cos\left(\sqrt{s \cdot \frac{1}{i^2}}\right)$, $2 < k \leq 25$
3	Вычислить сумму ряда: $\sum_{i=1}^k \left(\frac{1}{\sqrt{z \cdot i}} + \operatorname{tg}\left(\frac{k}{i}\right) \right)$, $k < 15$
4	Вычислить сумму ряда: $\sum_{i=1}^k \sqrt{a \cdot i} \cdot \sqrt{\frac{b}{i}}$, $k \leq 30$
5	Вычислить сумму ряда: $\sum_{i=1}^k \log_{10}\left(\sqrt{s \cdot \frac{1}{i^2}}\right)$, $k < 35$

6	Вычислить значение функции: $x(t,i) = \begin{cases} \sqrt{t}, & i=1 \\ \frac{1}{\sqrt{t}}, & i=2 \\ \sum_{k=1}^i k \cdot t, & i \neq 1, i \neq 2 \end{cases}$
7	Вычислить значение функции: $x(t,i) = \begin{cases} \ln(n), & i=1, 2 \\ \sum_{k=1}^i \frac{\sin(t)}{k}, & i > 2 \end{cases}$
8	Вычислить значение функции: $x(t,n) = \sum_{l=1}^n \begin{cases} \frac{1}{t} \cdot l, & l=1,3,5,\dots \\ \frac{1}{\sqrt{t}} \cdot l, & l=2,4,6,\dots \end{cases}$
9	Вычислить значение функции: $x(t) = \sum_{l=1}^t \begin{cases} \sqrt{t \cdot l}, & l=1,3,5,\dots \\ \frac{l}{\sqrt{t}}, & l=2,4,6,\dots \end{cases}$
10	Вычислить значение функции: $x(t,n) = \begin{cases} \sum_{i=1}^n t^2 \cdot i, & t < 0 \\ \sum_{i=1}^n \sqrt{t \cdot i}, & t \geq 0 \end{cases}$
11	Даны натуральное число n , действительные числа r, a_1, \dots, a_n ($n > 2$). Сколько среди точек $(a_1, a_2), (a_2, a_3), \dots, (a_{n-1}, a_n)$ таких, которые принадлежат кругу радиуса r с центром в начале координат? Вывести на экран координаты этих точек.
12	Даны натуральное число n , действительные числа x_1, \dots, x_n . Получить произведение членов последовательности x_1, \dots, x_n , кратных 3. Кроме того, получить сумму членов, принадлежащих отрезку $[3, 7]$, а также число таких членов.
13	Вычислить бесконечную сумму $\sum_{i=1}^{\infty} \frac{1}{i^2}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
14	Вычислить бесконечную сумму $\sum_{i=1}^{\infty} \frac{1}{i \cdot (i+1)}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
15	Вычислить бесконечную сумму $\sum_{i=1}^{\infty} \frac{(-1)^i}{i!}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое

	гаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
16	Вычислить бесконечную сумму $\sum_{i=0}^{\infty} \frac{(-2)^i}{i!(i+1)}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
17	Вычислить бесконечную сумму $\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i \cdot (i+1) \cdot (i+2)}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
18	Вычислить бесконечную сумму $\sum_{i=0}^{\infty} \frac{1}{4^i + 5^{i+2}}$ с заданной точностью ε ($\varepsilon > 0$). Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε . Это и все последующие слагаемые можно не учитывать.
19	Пусть $a_i = \frac{i-1}{i+1} + \sin \frac{(i-1)^3}{i+1}$, $i = 1, 2, \dots$. Дано натуральное n . Среди a_1, \dots, a_n найти все положительные числа, среди которых выбрать наименьшее положительное число.
20	Пусть $x_1 = 0.3$, $x_2 = -0.3$, $x_i = i + \sin(x_{i-2})$, $i = 2, 3, \dots$. Среди x_1, \dots, x_{100} найти ближайшее к какому-нибудь целому.
21	Даны натуральное число n , действительные числа y_1, \dots, y_n . Найти: $\max(z_1 , \dots, z_n)$, где $z_i = \begin{cases} y_i, & y_i \leq 2, \\ 0.5, & y_i > 2 \end{cases}$
22	Даны натуральное число n , действительные числа y_1, \dots, y_n . Найти: $\min(z_1 , \dots, z_n)$, где $z_i = \begin{cases} y_i, & y_i > 1, \\ 0.5, & y_i \leq 1 \end{cases}$
23	Даны натуральное число n , действительные числа y_1, \dots, y_n . Найти: $z_1 + \dots + z_n$, где $z_i = \begin{cases} y_i, & 0 < y_i < 10, \\ 1, & y_i \leq 0, y_i \geq 10 \end{cases}$
24	Даны натуральное число n , действительные числа y_1, \dots, y_n . Найти: $(\sqrt{z_1} - z_1)^2 + \dots + (\sqrt{z_n} - z_n)^2$, где $z_i = \begin{cases} y_i, & 0 < y_i \leq 15, \\ 2.7, & y_i \leq 0, y_i > 15 \end{cases}$
25	Даны натуральное число n , действительные числа y_1, \dots, y_n . Найти: $z_1^2 + \dots + z_n^2$, где $z_i = \begin{cases} y_i, & y_i < 1, \\ \frac{1}{y_i}, & y_i \geq 1 \end{cases}$
26	Даны целые числа a_1, \dots, a_{45} . Получить число отрицательных членов последовательности a_1, \dots, a_{35} и число нулевых членов по-

	последовательности a_{10}, \dots, a_{45} .
27	Даны натуральное число n , целые числа a, x_1, \dots, x_n . Если в последовательности x_1, \dots, x_n есть хотя бы один член, равный a , то получить сумму всех членов, следующих за первым таким членом; в противном случае ответом должно быть число -10.
28	Вычислить $\sum_{i=1}^{30} (a_i - b_i)^2$, где $a_i = \begin{cases} i, & i = 1, 3, 5, \dots \\ \frac{i}{2}, & i = 2, 4, 6, \dots \end{cases}$ $b_i = \begin{cases} i^2, & i = 1, 3, 5, \dots \\ i^3, & i = 2, 4, 6, \dots \end{cases}$
29	Пусть $x_0 = k, x_k = qx_{k-1} + b, (k=1, 2, \dots)$. Даны неотрицательное целое n , действительные a, b, c, d, q ($c < d$). Какие из x_0, \dots, x_n принадлежат интервалу (c, d) ?
30	Даны натуральное число n , целые числа a_1, \dots, a_n . Получить сумму положительных и число отрицательных членов этой последовательности.

ЛАБОРАТОРНАЯ РАБОТА № 3

Изучение функций в языке C++

Краткие теоретические сведения

Функции в C++ - аналоги процедур и функций в Паскале. Для использования функции необходимо сначала объявить прототип функции и определить функцию.

Пример использования функции в программе:

```
#include <stdio.h>
#include <math.h>
double func(double c); // Объявление прототипа функции
int main()
{
    double z=func(2.12345); // Вызов функции
    printf ("z=%20.26f",z) ;
    return 0;
}
double func(double c) // Определение функции
{
    return(sin(c)*cos(c));
} // Конец определения функции
```

3.1 Объявление прототипа функции

Для каждой функции в программе должен быть объявлен прототип до ее определения. Часто прототипы функций объявляют в заголовочном файле (типа .h или .hpp) и затем включают этот файл в программу с помощью директивы `#include "filename.h"`

Примеры объявления прототипов функций:

```
// 1. Объявление прототипов функций
void func1(void); // функция без параметров и не возвращает значение
void func2(int c); // функция с одним параметром целого типа и не возвращает значение
int func3(void); // функция без параметров, возвращает значение целого типа
double func4(double a, double b); // функция с двумя параметрами типа double и возвращает значение типа double
```

Функции `func1` и `func2` имеют тип `void`, являются аналогами процедур в Паскале и не возвращают значение.

3.2 Определение функции

Каждая объявленная функция должна быть определена в программе после объявления прототипа, но не внутри другой функции.

Пример определения функций:

```
void fund (void) {
```

```
//тело функции  
printf ("Вызов функции
```

Каждая функция типа, отличного от void, должна содержать хотя бы один оператор return, возвращающий значение функции:

```
double func4(double a, double b)  
{  
double d;  
d=a*b/(a+b);  
return d;  
}
```

3.3 Вызов функции

Примеры:

```
func1() ;  
double s=func4(2.32, 3.45);
```

3.4 Глобальные и локальные переменные. Область действия переменных

Переменные, объявленные вне функций, включая функцию main (), называются глобальными. Переменные, объявленные внутри функции, называются локальными по отношению к этой функции.

Примеры:

```
1.      #include <stdio.h>  
2.      void funct(void);  
3.      int global=9;  
4.      int main()  
5.      {  
6.          int locall;  
7.          locall=2;  
8.          for(int i=0;i<10;i++)  
9.              printf ("%d %d\n",locall*i,global*i);  
10.         funct();  
11.         return 0;  
12.     }  
13.     void funct(void)  
14.     {int local2=3;  
15.         printf ("%d",local2);  
16.     }
```

Область действия переменных: global - строки 3-16, locall - строки 6-12, local2 - строки 14-16, i - строки 8-9.

3.5 Передача параметров в функцию

Для передачи параметров в функцию можно реализовать с помощью:

а) явного указания параметров при вызове функции:

```
func(23.2, s);
```

б) глобальных переменных:

```
int func(void);
int global;
int main()
{
    global=23; int i=func();
}
int func(void)
{return (global*global*global);}
```

с) указателей, ссылок:

```
#include <stdio.h>
#include <math.h>
void func(double *p); //с помощью указателя
void func1 (double &r);
void func(double *p)
{
    *p= (*p)*(*p);
    printf("%f\n", (*p));
}
void func1(double &r); //с помощью ссылки
{r=sqrt(r);}
void main()
{
    double dd=3.14, // переменная
           *dp, // указатель
           &dr=dd; //ссылка на переменную dd
    dp=&dd; //инициализация - указатель указывает
           //на dd
    func (dp); //функция меняет значение переменной
           //dd, используя указатель
    printf ("%f\n%f\n", *dp, dd);
    func1(dr); //функция меняет значение переменной
           //dd, используя ссылку printf ("%f", dd);
}
```

```
Ответ: 9.859600
        9.859600
        9.859600
        3.140000
```

3.6 Передача результата из функции

Для передачи параметров в функцию можно реализовать с помощью:

- а) оператора **return**;
- б) глобальных переменных;
- с) указателей, ссылок;

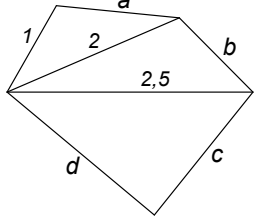
Задание на лабораторную работу № 3

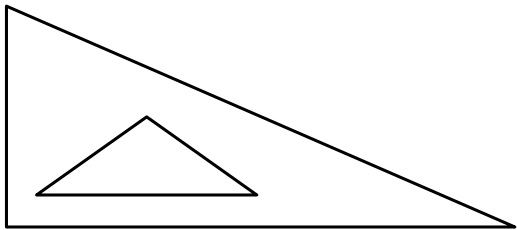
Модифицировать программу вычисления заданной функции из лабораторной работы №1, написав функции: ввода данных, вычисления результата по формуле, вывода данных;

Выполнить свой вариант задания из таблицы 3;

Таблица 3 – Варианты заданий для лабораторной работы № 3

Вариант	Задание
1	Дано натуральное число n ; найти $n!$. Использовать программу, включающую как рекурсивную, так и не рекурсивную функцию вычисления $n!$. Чем рекурсивная функция хуже не рекурсивной?
2	Числа Фибоначчи u_0, u_1, u_2, \dots определяются следующим образом: $u_0=0, u_1=1, u_n=u_{n-1}+u_{n-2} (n=2, 3, \dots)$. Написать программу вычисления u_n для данного неотрицательного целого n , включающую как рекурсивную функцию, которая основана на непосредственном использовании соотношения $u_n=u_{n-1}+u_{n-2}$, так и не рекурсивную функцию.
3	Даны натуральные числа a, c, m . Получить $f(n)$, где $f(n) = \begin{cases} n, & 0 \leq n \leq 9 \\ g(n)f(n-1-g(n))+n, & n > 9 \end{cases}$, $g(n)$ – остаток от деления $a(n+c)$ на 10. Использовать программу, включающую рекурсивную процедуру вычисления $f(n)$.
4	Даны неотрицательные целые числа n, m ; вычислить $A(n, m)$, где $A(n, m) = \begin{cases} m+1, & n=0 \\ A(n-1, 1), & n \neq 0, m=0 \\ A(n-1, A(n, m-1)), & n > 0, m > 0 \end{cases}$. Использовать программу, включающую рекурсивную функцию.
5	Составить процедуру, результатом работы которой является истинное значение, если символ, заданный при обращении к процедуре, – буква, и ложное значение в противном случае. Подсчитать, используя написанную функцию, количество букв и символов, не являющихся буквами в заданной строке.
6	Составить функцию, результатом работы которой является символ, заданный при обращении к функции, если этот символ не является буквой, и соответствующая строчная (малая) буква в противном случае. Обработать введенную строку, используя написанную функцию.
7	Составить функцию "сжатия" исходной последовательности символов: каждая последовательность, состоящая из нескольких вхождений одного и того же символа, заменяется на текст $x(k)$, где x -символ, а k -строка, являющаяся записью числа вхождений символа x в исходную последовательность. Обработать введенную строку, используя написанную функцию.
8	Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_N, y_N$. Найти периметр десятиугольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$. Написать функцию вычисления расстояния между двумя точками, заданными своими координатами.

9	<p>Даны действительные числа a, b, c, d. Найти площадь пятиугольника, изображенного на рисунке. Написать функцию вычисления площади треугольника по трем его сторонам.</p> 
10	<p>Даны натуральное число n, действительные числа $x_1, y_1, x_2, y_2, \dots, x_N, y_N$. Найти площадь n-угольника, вершины которого при некотором последовательном обходе имеют координаты $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$. Написать функцию вычисления площади треугольника по координатам его вершин.</p>
11	<p>Даны действительные числа s, t. Получить $h(s, t) + \max(h^2(s - t, st), h^4(s - t, s + t) + h(1, 1))$, где $h(a, b) = \frac{a}{1 + b^2} + \frac{b}{1 + a^2} - (a - b)^3$</p>
12	<p>Даны действительные числа $u_1, u_2, v_1, v_2, w_1, w_2$. Получить $3u + \frac{3uv}{4 + 2w - v} - 5$, где u, v, w – комплексные числа $u_1 + iu_2, v_1 + iv_2, w_1 + iw_2$. Написать функции вычисления арифметических операций над комплексными числами.</p>
13	<p>Даны действительные числа a_0, \dots, a_6. Получить для $x=1, 3, 4$ значения $p(x+1) - p(x)$, где $p(y) = a_6 y^6 + a_5 y^5 + \dots + a_0$.</p>
14	<p>Даны натуральное число n, действительное число x. Вычислить значение суммы ряда, используя рекурсивную функцию вычисления значения $n! \cdot \sum_{j=1}^n \frac{(2i)! + x }{(i^2)!}.$</p>
15	<p>Даны натуральное число n, действительное число x. Вычислить значение суммы ряда, используя рекурсивную функцию вычисления значения $\frac{1}{n!} \cdot \sum_{k=1}^n \left((-1)^k \cdot \frac{x^k}{(k! + 1)} \right).$</p>
16	<p>Дано натуральное число n. Вычислить значение суммы ряда, используя рекурсивную функцию вычисления значения $\frac{1}{n!} \cdot \sum_{i=1}^n ((-1)^i \cdot (2i - 1)!).$</p>
17	<p>Дано натуральное число n. Получить f_0, f_1, \dots, f_n, где $f_i = \frac{1}{i^2 + 1} + \frac{1}{i^2 + 2} + \dots + \frac{1}{i^2 + i + 1}$</p>
18	<p>Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_6, y_6$. Точки с координатами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, рассматриваются как вершины первого треугольника, точки с координатами $(x_4, y_4), (x_5, y_5), (x_6, y_6)$, - второго треугольника. Выяснить, верно ли, что первый треугольник целиком содержится во втором, и если да, определить площадь области, принадлежащей внешнему треугольнику и не принадлежащей внутреннему (на рисунке область заштрихована). (Определить процедуру, позволяющую выяснить, лежат ли две точки в одной полуплоскости относительно заданной прямой, процедуру вы-</p>

	<p>числения расстояния между двумя точками, а также процедуру вычисления площади треугольника по трем сторонам.)</p> 
19	Даны натуральное число n , символы s_1, \dots, s_n . Подсчитать сколько раз среди данных символов встречается символ $+$ и сколько раз символ $*$.
20	Даны действительные числа s, t . Получить $f(t, -2s, 1.17) + f(2.2, t, s-t)$, где $f(a, b, c) = \frac{2a - b - \sin c}{5 + c }$.
21	Даны действительные числа s, t . Получить $g(1.2, s) + g(t, s) - g(2s - 1, st)$, где $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$.
22	Даны натуральное число n , символы s_1, \dots, s_n . Подсчитать общее количество вхождений символов «+», «-», «*» в последовательность s_1, \dots, s_n .
23	Даны натуральное число n , символы s_1, \dots, s_n . Подсчитать наибольшее количество идущих подряд пробелов.
24	Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{50} \frac{1}{i + j^2}$.
25	Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{60} \sin(i^3 + j^4)$.
26	Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{100} \frac{j - i + 1}{i + j}$.
27	Вычислить $\sum_{i=1}^{100} \sum_{j=1}^i \frac{1}{2j + i}$.
28	Даны действительные числа $a_1, \dots, a_n, b_1, \dots, b_m$. В последовательности a_1, \dots, a_n и в последовательности b_1, \dots, b_m все члены, следующие за 3-м членом с наибольшим значением (за первым по порядку, если их несколько), заменить на 0.5.
29	Даны целые числа $a_1, \dots, a_n, b_1, \dots, b_m, k$. Если в последовательности a_1, \dots, a_n нет ни одного члена со значением k , то первый по порядку член этой последовательности, не меньший всех остальных членов, заменить на значение k . По такому же принципу преобразовать последовательность b_1, \dots, b_m применительно к значению 10.

30	<p>Даны натуральное число n, действительные числа $x, y, a_n, b_n, a_{n-i}, b_{n-i}, \dots, a_0, b_0$. Вычислить по схеме Горнера значение многочлена с комплексными коэффициентами</p> $(a_n + ib_n)(x + iy)^n + (a_{n-1} + ib_{n-1})(x + iy)^{n-1} + \dots + (a_0 + ib_0).$ <p>Определить процедуры выполнения арифметических операций над комплексными числами.</p>
----	--

ЛАБОРАТОРНАЯ РАБОТА № 4

Ввод, вывод. Файлы данных

Краткие теоретические сведения

4.1 Стандартные файлы ввода-вывода

Имя	Описание
Stdauх	Последовательный ввод-вывод
Stderr	Выходной поток ошибок
Stdin	Стандартный ввод
Stdout	Стандартный вывод
Stdprn	Вывод на принтер

4.2 Режимы доступа к файлам

Режим	Двоичный файл	Текстовый файл	Описание
"r"	"rb"	"rt"	Открыть существующий файл для чтения
"w"	"wb"	"wt"	Создать новый файл для записи. Пере-записывает существующий файл.
"a"	"ab"	"at"	Открыть существующий файл для чтения и записи в конец файла
"r+"	"r+b"	"r+t"	Открыть существующий файл для чтения и записи
"w+"	"w+b"	"w+t"	Создать новый файл для чтения и записи. <u>Перезаписывает существующий файл.</u>
"a+"	"a+b"	"a+t"	Открыть существующий файл для присоединения к его концу новой информации. Если файл не существует, создается новый файл с заданным именем.

4.3 Файловые функции

`FILE *fopen(const char *filename, const char *mode)`
Функция открывает существующий файл или создает новый.
Первый параметр - имя файла. Второй - режим доступа к файлу (см. 4.2).

`int fclose(FILE *stream)`

Закрывает файл. Возвращает значение 0, если файл закрыт успешно или EOF

`int fcloseall(void)` - Закрывает все открытые файлы.

`int fflush(FILE *stream)` - Выполнить довывод буферов потока в файл. При записи в файл, данные сначала попадают в специальный буфер. Если буфер переполняется, только тогда происходит сохранение данных в файле. Для принудительного сохранения используется функция `fflush`.

`int fgetc(FILE *stream)` - Читает символ из заданного потока. Если достигнут конец файла или произошла ошибка, возвращает значение EOF.

`char *fgets(char *s, int n, FILE *stream)` - Вводит строку символов из файла. Первый параметр `s` - указатель на строку, в которую производится ввод, `n` - количество символов в строке, `stream` – файловый поток, из которого производится ввод. В случае успеха возвращает строку, в случае ошибки или конца файла - NULL.

`int fputc(int c, FILE *stream)` - выводит символ `c` в поток `stream` (записывает в файл). В случае успеха возвращает `c`, в случае ошибки - EOF.

`int fputs(const char *s, FILE *stream)` - выводит строку `s` в поток. В случае успеха возвращает неотрицательное целое, в случае ошибки EOF.

Пример:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    char buffer[256];
    fp=fopen("file.dat", "r"); //открываем файл для чтения
    if(!fp)
        puts("Ошибка открытия файла ")
    else
    {
        while(fgets(buffer, 255, fp) != NULL) //пока не конец
                                                //файла, вводим
                                                строку
            fputs(buffer, stdout);           //выводим на экран
        fclose(fp);
    }
    return(0);
}
```

См. также функции **fprintf()**, **fwrite()**, **fread()**, **fseek()**, **fileno()**, **eof()**, **feof()**.

Задание на лабораторную работу № 4

Выполнить свой вариант задания из таблицы 4;

Таблица 4 – Варианты заданий для лабораторной работы № 4

Вариант	Задание
1	Дан файл f , компоненты которого являются действительными числами. Найти: а) сумму компонент файла f ; б) произведение компонент файла f .
2	Дан файл f , компоненты которого являются действительными числами. Найти: а) сумму квадратов компонент файла f ;

	б) модуль суммы и квадрат произведения компонент файла f ;
3	Дан файл f , компоненты которого являются действительными числами. Найти: а) наибольшее из значений компонент файла f ; б) наименьшее из значений компонент с четными номерами;
4	Дан файл f , компоненты которого являются действительными числами. Найти: а) наибольшее из значений модулей компонент с нечетными номерами; б) сумму наибольшего и наименьшего из значений компонент;
5	Дан файл f , компоненты которого являются целыми числами. Найти: а) количество четных чисел среди компонент; б) количество квадратов нечетных чисел среди компонент;
6	Последовательность x_1, x_2, \dots образована по закону $x_i = \frac{i - 0.1}{i^3 + tg 2i }, \quad i=1, 2, \dots$ Дано действительное $\varepsilon > 0$. Записать в файл h члены последовательности x_1, x_2, \dots , остановившись после первого члена, для которого выполнено $ x_i < \varepsilon$.
7	Дан символьный файл f . Получить копию файла в файле g .
8	Даны символьные файлы f1 и f2 . Переписать с сохранением порядка следования компоненты файла f1 в файл f2 , а компоненты файла f2 - в файл f1 . Использовать вспомогательный файл h .
9	Даны файлы f1, f2, f3, f4, f5 , компоненты которых являются действительными числами. Организовать обмен компонентами между файлами по схеме: f1 → f3 , f2 → f4 , f3 → f5 , f4 → f2 , f5 → f1 . Разрешается использовать только один вспомогательный файл h .
10	Дан символьный файл f . В файле f не менее двух компонент. Определить, являются ли два первых символа файла цифрами. Если да, то установить, является ли число, образованное этими цифрами, четным.
11	Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла f являющиеся четными числами.
12	Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла f делящиеся на 3 и не делящиеся на 7.
13	Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла являющиеся точными квадратами.
14	Дан файл f , компоненты которого u_0, u_1, \dots, u_n являются последовательными числами Фибоначчи. Получить в файле f последовательные числа Фибоначчи u_0, u_1, \dots, u_{n+1} .
15	Дан символьный файл f . Получить файл g , образованный из файла f заменой всех его прописных (больших) букв одноименными строчными (малыми).
16	Дан файл f , компоненты которого являются целыми числами. Записать в файл g все четные числа файла f , а в файл h - все нечетные. Порядок следования чисел сохраняется.

17	Даны символьные файлы f и g . Записать в файл h сначала компоненты файла f , затем - компоненты файла g с сохранением порядка.
18	Дан файл f , компоненты которого являются целыми числами. Получить файл g , образованный из файла f исключением повторных вхождений одного и того же числа.
19	Дан файл f , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h , переписать компоненты файла f в файл g так, чтобы в файле g сначала шли в положительные, потом отрицательные числа.
20	Дан файл f , компоненты которого являются целыми числами. Ни какая из компонент файла не равна нулю. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных и т.д. Переписать компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: пять положительных, пять отрицательных, пять положительных, пять отрицательных и т.д.
21	Дан файл f , компоненты которого являются целыми числами. Ни какая из компонент файла не равна нулю. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных и т.д. Переписать компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: двадцать положительных, двадцать отрицательных, двадцать положительных, двадцать отрицательных и т. д. (предполагается, что число компонент файла f делится на 40).
22	Дан символьный файл f . а) Подсчитать число вхождений в файл сочетаний <i>ab</i> . б) Определить входит ли в файл сочетание <i>abcdefgh</i> . Подсчитать число вхождений в файл каждой из букв <i>a, b, c, d, e, f</i> и вывести результат в виде таблицы: <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;">$a-N_a$</div> <div style="text-align: center;">$b-N_b$</div> <div style="text-align: center;">$c-N_c$</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <div style="text-align: center;">$d-N_d$</div> <div style="text-align: center;">$e-N_e$</div> <div style="text-align: center;">$f-N_f$</div> </div>
23	Даны символьные файлы f и g . Определить, совпадают ли компоненты файла f с компонентами файла g . Если нет, то получить номер первой компоненты, в которой f и g отличаются между собой. В случае, когда один из файлов имеет <i>n</i> компонент ($n > 0$) и повторяет начало другого (более длинного) файла, ответом должно быть число $n+1$.
24	Дан символьный файл f Записать в файл g с сохранением порядка следования те символы файла f , которым в этом файле предшествует буква <i>a</i> .
25	Дан символьный файл f . Записать в файл g с сохранением порядка следования те символы файла f , вслед за которым в этом файле идет буква <i>a</i> .
26	Дан символьный файл f . Считая, что количество символов в слове не превосходит двадцати определить, сколько в файле f имеется слов, состоящих из одного, двух, трех и т.д. символов.
27	Дан символьный файл f . Считая, что количество символов в слове не превосходит двадцати получить гистограмму (столбча-

	тую диаграмму) длин всех слов файла f .
28	Дан символьный файл f , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия_имя_отчество, фамилия_имя_отчество,... Записать эти сведения в файле g , используя образцы: имя_отчество_фамилия, имя_отчество_фамилия, имя_отчество_фамилия.
29	Дан символьный файл f , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия_имя_отчество, фамилия_имя_отчество,... Записать эти сведения в файле g , используя образцы: фамилия и.о., фамилия и.о., фамилия и.о.,...
30	Дан файл f , содержащий различные даты. Каждая дата - это число, месяц и год. Найти: а) год с наименьшим номером; б) все весенние даты; в) самую позднюю дату.

ЛАБОРАТОРНАЯ РАБОТА № 5

Работа с массивами

Краткие теоретические сведения

5.1 Объявление массивов

В массивах объединяются элементы одного и того же типа.

Объявление одномерного массива:

```
int array[100];
```

Объявление двумерного массива 5x10 элементов `int arr1[5][10];`

В языке C++ массивы имеют следующие особенности:

- индекс элементов массива изменяется от 0 до N-1;
- выход значения индекса за пределы диапазона 0...N-1 не контролируется. Если значение индекса выйдет за пределы диапазона 0...N-1, произойдет доступ к области памяти, не принадлежащей массиву, что может привести к серьезным ошибкам.

Индексами могут быть любые целые выражения - константы, переменные, значения функций.

5.2 Инициализация массивов

```
int digit [10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

Следующая строка инициализирует массив и заставляет компилятор вычислить размер массива автоматически:

```
int digit [] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }; Сле-
```

дующая строка инициализирует часть массива:

```
int digit[10] = { 0, 1, 2, 3, 4, 5 }; Ини-
```

циализация многомерных массивов:

```
int array[3][4] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 }
```

В данном примере `array[0][0]=0,..., array[0][3]=3, array[1][0]=4` и т.д.

5.3. Использование функции **sizeof**

Если **Array** - имя массива, то выражение **sizeof(Array)** равно числу байтов, занимаемых массивом, а выражение **sizeof(Array[0])** - числу байтов, занимаемых одним элементом.

Передача массивов функциям.

Если **Array** - имя массива, то **Array[]** - указатель на массив. Для передачи массива функциям необходимо передать в функцию указатель на массив и количество элементов массива.

Для передачи многомерного массива необходимо передать в функцию как количество строк, так и количество столбцов, чтобы указать компилятору, какую структуру имеет массив:

```
void fun(int data[][4], int rows);
```

В следующей программе приведены примеры передачи одно- и двумерных массивов в функцию:

```
#include <iostream.h>
int sum (int array[], int n);
int sum1 (int array1[][4], int m, int n);
int sum (int array[], int n)
```

```
{
int s=0;
for(int i1=0; i1<n; i1++)
s+=array[i1];
return s;
}
int sum1 (int array[][4], int m)
int s=0;
for(int i2=0; i2<m; i
for(int j2=0; j2<4; J2++)
s+=array[i2j return s;

void main ()
{
int array[10] = {0,1,2,3,4,5,6,7,8,9} ;
int array1[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11} ;
for (int i=0;i<3;i++)
for (int j=0;j<4;j++)
cout << "array1 ["<<i<<" ["<<j<<" "«array1 [i] [j]«"\n";
int s1=sum(array,10); int s2=sum1(array1,3); cout <<
"\ns1="<<s1<<"\n"; cout << "sum2 = "<<s2;
```

Результат выполнения программы:

```
array1[0][0]0
array1[0][1]1
array1[0][2]2
array1[0][3]3
array1[1][0]4
array1[1][1]5
array1[1][2]6
array1[1][3]7
array1[2][0]8
array1[2][1]9
array1[2][2]10
array1[2][3]11

sum1=45
sum2=66
```

Задание на лабораторную работу № 5

Каждое задание решается с применением функций, в которые передаются массивы. Исходные данные вводить из файла, с клавиатуры или генерировать с помощью функции *random*;

Выполнить свой вариант задания из таблицы 5;

Таблица 5 – Варианты заданий для лабораторной работы № 5

Вариант	Задание
1	Дана целочисленная квадратная матрица порядка 8. Найти наименьшее из значений элементов столбца, который обладает наибольшей суммой модулей элементов. Если таких столбцов не-

	сколько, то взять первый из них.
2	Даны натуральное число n , целочисленная квадратная матрица порядка n . Получить b_1, \dots, b_n , где b_i это: а) Наименьшее из значений элементов, находящихся в начале i -й строки матрицы до элемента, принадлежащего главной диагонали, включительно; б) Значение первого по порядку положительного элемента i -й строки (если таких элементов нет, то принять $b_i=1$).
3	Даны натуральное число n , целочисленная квадратная матрица порядка n . Получить b_1, \dots, b_n , где b_i это: а) Сумма элементов, расположенных за первым отрицательным элементом в i -й строке (если все элементы строки неотрицательны, принять $b_i=100$); б) Сумма элементов, предшествующих последнему отрицательному элементу i -й строки (если все элементы строки неотрицательны, то принять $b_i=-1$).
4	Дана целочисленная матрица порядка n . Найти номера строк: а) все элементы которых - нули; б) элементы в каждой из них одинаковы.
5	Дана целочисленная матрица порядка n . Найти номера строк: а) все элементы которых четны; б) элементы каждой из них образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую).
6	Дана действительная квадратная матрица порядка 10. В строках с отрицательным элементом на главной диагонали найти: а) сумму всех элементов; б) наибольший из всех элементов.
7	Дана действительная квадратная матрица порядка n . Рассмотрим те элементы, которые расположены в строках, начинающихся с отрицательного элемента. Найти суммы тех из них, которые расположены соответственно ниже, выше и на главной диагонали.
8	Дана действительная квадратная матрица порядка 9. Получить целочисленную квадратную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше элемента, расположенного в его строке на главной диагонали, и равен нулю в противном случае.
9	Дана действительная квадратная матрица порядка n . Получить $x_1x_n + x_2x_{n-1} + \dots + x_nx_1$; где x_k - наибольшее значение элементов k -ой строки данной матрицы.
10	Даны натуральное число $n > 2$, действительная квадратная матрица порядка n . Построить последовательность b_1, \dots, b_n из нулей и единиц, в которой $b_i=1$ тогда и только тогда, когда элементы i -й строки матрицы образуют возрастающую последовательность.
11	Даны натуральное число $n > 2$, действительная квадратная матрица порядка n . Построить последовательность b_1, \dots, b_n из нулей и единиц, в которой $b_i=1$ тогда и только тогда, когда элементы i -й строки матрицы образуют возрастающую или убывающую последовательность.
12	Дана действительная квадратная матрица порядка n . Построить

	последовательность действительных чисел a_1, \dots, a_n по правилу: если в i -й строке матрицы элемент, принадлежащий главной диагонали, отрицателен, то a_i равно сумме элементов i -й строки, предшествующей первому отрицательному элементу; в противном случае a_i равно сумме последних элементов i -й строки, начиная с первого по порядку неотрицательного элемента.
13	Дана целочисленная квадратная матрица порядка 15. Выяснить, имеются ли в матрице ненулевые элементы, и если имеются, то указать индексы всех ненулевых элементов.
14	Даны натуральное число n , действительная квадратная матрица порядка n . Построить последовательность b_1, \dots, b_n из нулей и единиц, в которой $b_i=1$ тогда и только тогда, когда в i -й строке матрицы есть хотя бы один отрицательный элемент.
15	С помощью $[X_{ij}]$, где $i=1, 2; j=1, \dots, n$ действительной матрицы на плоскости задано n точек так, что $X_{1,j}, X_{2,j}$ - координаты j -й точки. Точки попарно соединены отрезками. Найти длину наибольшего отрезка.
16	Дана действительная матрица размера $m \times n$, в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.
17	Дана действительная квадратная матрица порядка 12. Заменить нулями все ее элементы, расположенные на главной диагонали и выше нее.
18	Дана действительная квадратная матрица размера 6×9 . Найти среднее арифметическое наибольшего и наименьшего значений ее элементов.
19	Дана действительная матрица размера $m \times n$. Найти сумму наибольших значений элементов ее строк.
20	В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.
21	Даны натуральное число n , действительная квадратная матрица порядка n , действительные a_1, \dots, a_n+5 . Элементы последовательности a_1, \dots, a_n+5 домножить на 10, если наибольший элемент матрицы (в предположении, что такой элемент единственный) находится на главной диагонали, и на 0.5 в противном случае.
22	Дана действительная матрица размера $m \times n$, все элементы которой различны. В каждой строке выбирается элемент с наименьшим показателем, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.
23	Дана действительная квадратная матрица порядка 9. Вычислить сумму тех из ее элементов, расположенных на главной диагонали и выше нее, которые превосходят по величине все элементы, расположенные ниже главной диагонали. Если на главной диагонали и выше ее нет элементов с указанным свойством, то ответом должно служить сообщение об этом.

24	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $\sum_{j=1}^n a_{i,j}^2$.
25	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $\sum_{j=1}^n (-1)^{i+j} a_{i,j}$.
26	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $\prod_j a_{i,j}$.
27	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $\sum_{j=1}^n a_{i,j} $.
28	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $\prod_j a_{i,j}$, для всех таких j , что $1 \leq a_{i,j} \leq n$.
29	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $[\max_j a_{i,j}]_{j=1,\dots,n}$.
30	Дана целочисленная матрица $[a_{i,j}]_{i,j=1,\dots,n}$. Получить b_1, \dots, b_n , где b_i и это $[\min_j a_{i,j}]_{j=1,\dots,n}$.
31	Таблица футбольного чемпионата задана квадратной матрицей порядка n , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2- выигрыш, 1 - ничья, 0 - проигрыш). Найти число команд, имеющих больше побед, чем поражений.
32	Таблица футбольного чемпионата задана квадратной матрицей порядка n , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2- выигрыш, 1 - ничья, 0 - проигрыш). Определить номера команд, прошедших чемпионат без поражений.
33	Таблица футбольного чемпионата задана квадратной матрицей порядка n , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2- выигрыш, 1 - ничья, 0 - проигрыш). Выяснить, имеется ли хотя бы одна команда, выигравшая более половины игр.

ЛАБОРАТОРНАЯ РАБОТА № 6

Работа со строками

Краткие теоретические сведения

Строка представляет собой массив значений типа **char**, который заканчивается нулевым байтом. Нулевой байт - не символ '0' (ноль), а байт, равный нулю (0x00). Если строка заполнена не полностью, нулевой байт указывает на конец строки, а байты, следующие за концом строки, ничем не заполняются (содержат случайные значения).

Строковые литералы - строки, вводимые непосредственно в текст программы:

```
char str1[]="My Program";
```

"My Program" - строковый литерал.

Для работы со строками используются строковые переменные и строковые указатели, str1 в предыдущем примере является строковой переменной.

Строковые указатели будут рассмотрены в лабораторной работе 7.

Нулевая строка - строка, начинающаяся с нулевого символа – признака конца строки (пустая строка). Длина нулевой строки равна нулю.

Функции работы со строками

Ввод строки	gets(string)
Вывод строки	puts(string)
Определение длины строки	strlen(string)
Копирование строк	strcpy(string1, string2)
Дублирование строк	str1=strdup(str2)
Сравнение строк	strcmp(string1, string2)
Конкатенация (объединение) строк	strcat(string1, string2)
Поиск символов в строке	strchr(string, chr)
Поиск подстроки в строке	strstr(string1, string2)

Другие функции - см. Том Сван. Освоение Borland C++4.5. Энциклопедия функций. Большинство имен строковых функций начинается на str...

Задание на лабораторную работу № 6

Исходные данные для задач вводятся с клавиатуры или из файла. Результат выводится на экран или записывается в файл;

Выполнить свой вариант задания из таблицы 6

Таблица 6 – Варианты заданий для лабораторной работы № 6

Вариант	Задание
1	Дан символьный файл f , содержащий произвольный текст, длиной более 5000 слов. Слова в тексте разделены пробелами и знаками препинания. Получить 100 наиболее часто встречающихся слов и

	число их появлений. Решить задачу: а) Без ограничения на длины слов; б) Предполагая, что любое слово текста состоит не более чем из 16 букв
2	Дан символьный файл f . Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Удалить из файла все однобуквенные слова и лишние пробелы. Результат записать в файл g .
3	Дан символьный файл f . Записать в файл g с сохранением порядка следования те символы файла f : а) Которым в этом файле предшествует буква <i>a</i> ; б) Вслед за которым в этом файле идет буква <i>a</i> .
4	Дан символьный файл f . Найти самое длинное слово среди слов, вторая буква которых есть <i>e</i> ; если слов с наибольшей длиной несколько, то найти последнее. Если таких слов нет вообще, то сообщить об этом. Решить эту задачу: а) Полагая, что слова состоят не более чем из 10 символов; б) Без ограничения на число символов в строке.
5	Дан символьный файл f . Считая, что количество символов в строке не превосходит двадцати определить, сколько в файле f имеется слов, состоящих из одного, двух, трех и т.д. символов.
6	Дан символьный файл f , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия_имя_отчество, фамилия_имя_отчество,... Записать эти сведения в файле g , используя образцы: имя отчество фамилия, имя отчество фамилия,...
7	Дан символьный файл f , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия_имя_отчество, фамилия_имя_отчество,... Записать эти сведения в файле g , используя образцы: фамилия_и.о., фамилия_и.о., фамилия_и.о.,...
8	Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл f , содержащий сведения об учениках школы. Выяснить, имеются ли в школе однофамильцы.
9	Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл f , содержащий сведения об учениках школы. Получить список учеников данного класса по следующим образцам: фамилия имя, фамилия и., и. фамилия.
10	Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f , содержащий сведения о нескольких автомобилях. Найти фамилии владельцев и номера автомобилей данной марки.
11	Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f , содержащий сведения о нескольких автомобилях. Найти количество автомобилей данной марки.
12	Даны два символьных файла f1 и f2 Файл f1 содержит произвольный текст. Слова в тексте разделены пробелами и знаками препи-

	<p>нения. Файл f2 содержит не более 40 слов, которые разделены запятыми. Эти слова образуют пары: каждое первое слово считается заменяемым, каждое второе слово - заменяющим. Найти в файле f1 все заменяемые слова и заменить на соответствующие заменяющие. Результат поместить в файле g.</p>
13	<p>Дан файл f, который содержит номера телефонов сотрудников учреждения; указывается фамилия сотрудника, его инициалы и номер телефона. Найти телефон сотрудника по его фамилии и инициалам.</p>
14	<p>Дан файл f, содержащий об экспортируемых товарах: указывается наименование товара, страна, импортирующая товар, и объем поставляемой партии в штуках. Найти страны, в которые экспортируется данный товар, и общий объем экспорта.</p>
15	<p>Дан файл f, содержащий сведения о веществах: указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор):</p> <p>а) Найти удельные веса и названия всех полупроводников;</p> <p>б) Выбрать данные о проводниках и упорядочить их по убыванию удельных весов.</p>
16	<p>Дан текст. Если в тексте нет символа «*», то оставить этот текст без изменения, иначе каждую из малых латинских букв, предшествующих первому вхождению символа «*», заменить на цифру 3.</p>
17	<p>Дан текст. Если в тексте нет символа «+», то оставить текст без изменения, иначе каждую из цифр, предшествующую первому вхождению символа «+», заменить символом «-».</p>
18	<p>Дан текст. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть, как и прежде словами. Найти все слова, содержащие наибольшее количество гласных латинских букв (<i>a, e, i, o, u</i>).</p>
19	<p>Дан текст. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть, как и прежде словами. Найти все слова, в которых доля букв <i>a, b</i> максимальна.</p>
20	<p>Дан текст. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть, как и прежде словами. В тех словах, которые канчиваются сочетанием <i>ing</i>, заменить это окончание на <i>ed</i>.</p>
21	<p>Дано натуральное число <i>n</i>, равное выраженной в копейках цене некоторого товара, например 317, 5005, 100 и т.д. Выразить цену в рублях и копейках, например 3 руб. 17 коп., 50 руб. 05 коп., 1 руб. 00 коп. и т.д. (Число копеек записывается всегда двумя цифрами).</p>
22	<p>Дан текст, каждый символ которого может быть малой буквой, цифрой или одним из знаков «+», «-», «*». Группой букв будем называть такую совокупность последовательно расположенных букв, которой непосредственно не предшествует и за которой непосредственно не следует буква. Аналогично определим группу цифр и группу знаков.</p> <p>а) Выяснить, встречается ли в данном тексте группа букв <i>one</i>;</p> <p>б) Выяснить, верно ли, что в данном тексте больше групп букв, чем групп знаков.</p>

23	<p>Условие - смотри вариант 22.</p> <p>а) Если в данном тексте имеется не менее двух групп букв, то каждый знак «+», встречающийся между двумя первыми по порядку группами букв, заменить цифрой 1, знак - заменить цифрой 2, а знак «*» - цифрой 3. Иначе оставить текст без изменения;</p> <p>б) Подсчитать количество вхождений буквы <i>f</i> в первые три группы букв (в предположении, что текст содержит не менее трех групп букв).</p>
24	<p>Условие - смотри вариант 22.</p> <p>а) Найти число таких групп букв, которые начинаются и кончаются одной и той же буквой;</p> <p>б) Найти все такие группы букв, в которые буква <i>a</i> входит не менее двух раз.</p>
25	<p>Для большинства существительных, оканчивающихся на — <i>опок</i> и — <i>елок</i>, множественное число образуется от другой основы. Как правило, это происходит по образцу: цыпленок - цыплята, мышонок - мышата и т.д. (в новой основе перед последней буквой <i>т</i> пишется <i>а</i> или <i>я</i> в зависимости от предыдущей буквы: если это шипящая, то <i>а</i>, если иначе - <i>я</i>). Имеются слова-исключения, из которых укажем следующие: ребенок (дети), бесенок (бесенята), опенок (опята), звонок (звонки), позвонок (позвонки), подонок (подонки), колонок (колонки), жаворонок (жаворонки), бочонок (бочонки). Есть еще ряд малоупотребительных слов-исключений, которые мы не рассматриваем. Дан текст, среди символов которого имеется пробел. Группа символов, предшествующая первому пробелу, представляет собой русское слово, оканчивающееся на <i>-опок</i> или <i>-елок</i>. Получить это слово во множественном числе.</p>
26	<p>Даны натуральное число <i>n</i>, символ <i>s</i> ($n < 1000$, <i>s</i> - одна из букв <i>и, р, д, в, т, н</i>, указывающая падеж - именительный, родительный, дательный, винительный, творительный, предложный). Записать количественное числительное, обозначающее <i>n</i>, в соответственном падеже.</p>
27	<p>Даны символьные файлы f и g. Определить, совпадают ли компоненты файла f с компонентами файла g. Если нет, то получить номер первой компоненты, в которой файлы f и g отличаются между собой. В случае, когда один из файлов имеет <i>n</i> компонент ($n > 0$) и повторяет начало другого (более длинного) файла, ответом должно быть число $n+1$.</p>
28	<p>Дан символьный файл f. Предполагается, что длина одного слова не превосходит десяти и что число слов делится на 100. Подготовить файл для печати слов в две колонки по пятьдесят строк на странице. Слова должны быть размещены в файле f1 в следующем порядке: 1-е слово, 51-е слово, 2-е слово, 52-е слово, ..., 50-е слово, 100-е слово, затем (следующая страница) 101-е слово, 151-е слово, ..., 150-е слово, 200-е слово и т.д.</p>
29	<p>Дан символьный файл f. Подсчитать число вхождений в файл сочетаний <i>ab</i>.</p>
30	<p>Дан символьный файл f. Определить, входит ли в файл сочетание <i>abcdefgh</i>.</p>

31	Дан текстовый файл, содержащий программу на языке Паскаль. Проверить эту программу на несоответствие числа открывающих и закрывающих круглых скобок. Считать, что каждый оператор программы а) Занимает не более одной строки файла f ; б) Может занимать произвольное число строк файла f .
32	Дан текстовый файл f . Записать в перевернутом виде строки файла f в файл g . Порядок строк в файле должен а) совпадать с порядком исходных строк в файле f ; б) Быть обратным по отношению к порядку строк исходного файла.
33	Даны символьные файлы f и g . Записать в файл h все начальные совпадающие компоненты файлов f и g .

ЛАБОРАТОРНАЯ РАБОТА № 7

Указатели и ссылки

Краткие теоретические сведения

7.1 Указатели

Указатели делятся на две основные категории: указатели объектов и указатели функций. Указатели обоих типов представляют собой специальные объекты, хранящие адреса памяти.

Объявление указателя всегда должно устанавливать его на некоторый конкретный тип, даже если этот тип `void` (что фактически означает указатель на любой тип). Однако, уже после объявления указатель обычно может быть переназначен на объект другого типа.

Примеры объявления указателей:

```
int i;           // i это целое;
int * pi;        // pi это указатель на i
                // (неинициализированный)
int * const cp = &i; // cp это указатель-константа на int const
int ci = 7;      // ci это константа int const
int * pci;       // pci это указатель на константу ci
const int * const cpc = &ci; // cpc это указатель-константа на константу int
```

7.2 Ссылки

Типы ссылок C++ тесно связаны с типами указателей. Типы ссылок C++ служат для создания алиасов объектов и позволяют передачу аргументов функциям по ссылке. Традиционно C передает аргументы только по значению. C++ позволяет передавать аргументы как по значению, так и по ссылке.

Для объявления ссылок вне функции может быть использован декларатор ссылки:

```
int i = 0;
int &ir = i; // ir является алиасом i
ir = 2;      // то же, что i = 2
```

В данном примере создается именуемое значение `ir`, являющееся алиасом `i`, при условии, что инициализатор имеет тот же тип, что и ссылка. Выполнение операций с `ir` имеет тот же результат, что и выполнение их с `i`. Например, `ir = 2` присваивает 2 переменной `i`, а `&ir` возвращает адрес `i`.

Декларатор ссылки может также быть использован для объявления в функции параметров типа ссылки:

```
void func1 (int i);
void func2 (int &ir); // ir имеет тип "ссылка на int"
int sum=3;
func1(sum);          // sum передается по значению
```

`func2(sum)` *// sum передается по ссылке*

Переданный по ссылке аргумент `sum` может быть изменен прямо в `func2`. Напротив, `func1` получает только копию аргумента `sum` (переданного по значению), поэтому сама переменная `sum` функцией `func1` изменена быть не может.

Задание на лабораторную работу № 7

Модифицировать программу к любой из лабораторных работ с использованием указателей и ссылок для доступа к переменным и элементам массивов;

ЛАБОРАТОРНАЯ РАБОТА № 8

Работа с памятью

Краткие теоретические сведения

Управление оперативной памятью осуществляется с помощью библиотечных функций **malloc**, **calloc** **free**. Функции **malloc**, **calloc** осуществляют выделение памяти для данных, а **free** - ее освобождение. Описание этих функций находится в библиотечном файле **stdio.h**.

8.1 Функция malloc()

Описание: **void *malloc(unsigned size)**

Функция **malloc** возвращает в качестве своего значения указатель на область памяти, начинающуюся на границе слова. Размер этой области в байтах определяется с помощью аргумента. Если выделение памяти невозможно, то результатом будет пустой адресный указатель.

Например:

```
int *x;  
x = (int *) malloc( sizeof(int) ); //выделение памяти для типа int
```

8.2 Функция calloc()

Описание: **void *calloc(unsigned count, unsigned size)**

Функция имеет два аргумента, которые должны быть целыми без знака. Первый аргумент **count** определяет количество требуемых ячеек памяти. Второй аргумент **size** - размер каждой ячейки в байтах

Выполнение функции приводит к увеличению объема оперативной памяти на величину **count * size** (**count** умножить на **size**). Результатом функции является указание адреса увеличенного объема памяти.

Кроме того, функция **calloc()** предварительно обнуляет всю выделенную память.

Если увеличение объема не происходит, то результатом функции будет указание на пустой адрес.

Пример:

```
char *calloc();  
long *newmem;  
  
newmem=(long*) calloc(100, sizeof(long));
```

Здесь **calloc()**, как и прежде **malloc()** объявлен типа **char**. Для выделения памяти под переменную типа **long** используется оператор преобразование типа (**long ***). Применение **sizeof(long)** (вместо 4 - размер в байтах типа **long**)делает программу более мобильной. Она будет нормально работать в тех системах, где тип **long** имеет размер отличный от 4 байт.

8.3 Функция `free()`

Объявление: **`void free(void *ptr)`**

Функция **`free`** обеспечивает в результате своего выполнения освобождение выделенной ранее с помощью функции **`calloc`** области памяти, определяемой с помощью аргумента *ptr*.

8.4 Операторы *new*, *delete*

Функции **`malloc ()`** и **`free()`** выделяют и освобождают блоки памяти в “куче”. Такой вид памяти обычно называют *динамической памятью*, потому что она создается и удаляется при выполнении программы. Во время работы с объектами часто необходимо создавать элементы динамически. Создатели языка C++ придавали этому такое значение, что добавили в язык три встроенных оператора:

- `new`
- `delete`
- `delete []`

Оператор `new` по своему назначению близок к функции **`malloc ()`**, но это оператор, а не функция. Этот оператор является частью самого языка, как и **`delete`**, поэтому, чтобы его использовать, не нужно включать никаких заголовочных файлов.

Задание на лабораторную работу № 8

Модифицировать программу к любой из лабораторных работ с использованием динамического выделения памяти для переменных и массивов.