




ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра «Информационные технологии»

**Методические указания**  
к выполнению лабораторных работ  
по дисциплине  
**«Технология  
программирования»**

Авторы  
Рашидова Е.В.,  
Колесникова О.В.



Ростов-на-Дону, 2023

## Аннотация

Методические указания предназначены для организации лабораторных и самостоятельных работ студентов направления 09.03.02 – Информационные системы и технологии всех форм обучения. В указаниях собраны и систематизированы задачи по основным разделам курса. Задачи полезны для освоения и отработки основных технологий программирования на базе языка C/C++.

## Авторы



Доцент, к.ф.-м. кафедры  
Информационные технологии  
Рашидова Е.В.



Старший преподаватель  
кафедры Информационные  
технологии  
Колесникова О.В.



## Оглавление

Лабораторная работа № 1. «ЧИСЛОВЫЙ ТИП. ОПЕРАТОР ПРИСВАИВАНИЯ» .....	4
Лабораторная работа № 2. «ПРЕОБРАЗОВАНИЕ ТИПОВ» ...	6
Лабораторная работа № 3. «ОПЕРАТОРЫ СРАВНЕНИЯ. ЛОГИЧЕСКИЕ ОПЕРАТОРЫ» .....	8
Лабораторная работа № 4. «ПОБИТОВЫЕ ОПЕРАЦИИ» .....	10
Лабораторная работа № 5. «УСЛОВНЫЙ ОПЕРАТОР И ОПЕРАТОР SWITCH» .....	11
Лабораторная работа № 6. «ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ ЯЗЫКА СИ» .....	14
Лабораторная работа № 7. «ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ ЯЗЫКА СИ» .....	17
Лабораторная работа № 8. «ОДНОМЕРНЫЕ МАССИВЫ» ..	19
Лабораторная работа № 9. «СТРОКИ» .....	20
Лабораторная работа № 10. «Стандартная библиотека Си для работы с символьными строками» .....	21
Список литературы .....	23

## ЛАБОРАТОРНАЯ РАБОТА № 1. «ЧИСЛОВОЙ ТИП. ОПЕРАТОР ПРИСВАИВАНИЯ»

Цель работы: Изучения числового типа данных, оператора присваивания.

Форма отчета: демонстрация выполненной работы преподавателю.

1.1. Для верно записанных целых констант определить их значение, тип и систему счисления.

- a) 123;                      b) 1E6;                      c) 123456789LU;                      d) -5;  
 e) 0XFUL;                      f) 058;                      g) 0X-1AD;                      h) 00123;  
 i) 0xfffffL;                      j) 01A;                      k) 0731UL;                      l) +0xaf;                      m) 0X0

1.2. Указать верно записанные символьные константы и способ их представления.

- a) '0';                      b) '\x7';                      c) '\122';                      d) -'x';  
 e) "x";                      f) 'a'U;                      g) '\n';                      h) 97

1.3. Для верно записанных вещественных констант определить их значение, тип.

- a) 1.71;                      b) 1E-6;                      c) 0.314159E1F;                      d) .005;  
 e) 0051E-04;                      f) 5.E+2;                      g) 0e0;                      h) 0x1A1.5;  
 i) 05.5;                      j) 0;                      k) 0X1E6;                      l) 0F;  
 m) 1234.56789L1                      n) 1.0E-10D;                      o) 3.1415U;                      p) 1e-2f;  
 q) -12.3E-6;                      r) +10e6;                      s) 123456L;                      t) E-6

1.4. Для верно записанных выражений вычислить их значения (операции + - \* / % =):

`int a,b,c,d,e;`

`a=2; b=13; c=7; d=19; e=-4;`

- a) `b/a/c;`                      b) `d/a%c;`                      c) `c%d-e;`                      d) `-e%a+b/a*-5+5;`  
 e) `b%e;`                      f) `7-d%+(3-a);`                      g) `b%-e*c;`                      h) `9/c- -20/d`

1.5. Пусть `int a,c; c=5;`

Значение переменной `s` увеличить на 1, переменной `a` присвоить значение, равное удвоенному значению переменной `s`. Укажите верные варианты решения задачи:

- a) `s++;` `a=2*s;`                      b) `a=2*s++;`                      c) `s+=1;a=s+c;`                      d) `a=s++ +c;`  
 e) `++s;` `a=s+c;`                      f) `a=++c+c;`                      g) `a=s+=1+c;`                      h) `a=(s+=1)+c`

1.6. Пусть `int a,c; c=5;` значение переменной `s` уменьшить на 1; переменной `a` присвоить значение, равное частному от деления переменной `s` на 2».

- a) `--s;` `a=c/2;`                      b) `a=-c/2;`                      c) `c-=1;a=c%2;`                      d) `a=c--/2;`  
 e) `a=c-=1/2;`                      f) `a=(c-=1)/2.0`                      g) `a=(c-=1)/2;`                      h) `a=(c=c-1)/2;`

1.7. Эквивалентны ли выражения?

a)  $E1 \text{ op} = E2$  и  $E1 = E1 \text{ op} E2$

b)  $E1 \text{ op} = E2$  и  $E1 = E1 \text{ op} (E2)$

Замечание: здесь  $E1, E2$  - выражения допустимого в этом случае типа;  $\text{op}$  - операция (одна из  $+ - * / \% \gg \ll \& \wedge |$ ).

1.8. Для верно записанных выражений вычислить их значения, определить тип результата:

`int i = 1, j = 2, k = -7, m = 0;`

a)  $k * = 3 + j$ ;                      b)  $i += ++j + 3$ ;                      c)  $k \% = m = 1 + k / 2$ ;

d)  $1 + 3 * k += 7 / 5$ ;                      e)  $1 + 3 * (k += 7) / 5$ ;

1.9. Для верно записанных выражений вычислить их значения (операции  $+ - * / ++ --$  операции присваивания):

`int a, b, c; a = 2; b = 6; c = 3;`

a)  $--a$ ;                      b)  $--a$ ;                      c)  $b--a$ ;                      d)  $a += a++$ ;

e)  $++b / a++ * --c$ ;                      f)  $a--b$ ;                      g)  $a--b$ ;                      h)  $a++ = b$ ;

i)  $a = a++$ ;                      j)  $b++ / ++a * c--$ ;                      k)  $--a$ ;                      l)  $a--c$ ;

m)  $a++ = a$ ;                      n)  $++a = b$ ;                      o)  $a = (b + 1)++$

1.10. Допустимо ли в Си? Если "да" - опишите семантику этих действий; если "нет" - объясните почему.

`int x = 5;`  
`++ x = 10;`  
`printf ("%d\n", x);`

1.11. Дано вещественное число  $x$ . Не пользуясь никакими операциями, кроме умножения, сложения и вычитания, вычислить  $2x^4 - 3x^3 + 4x^2 - 5x + 6$ . Разрешается использовать не более четырех умножений и четырех сложений и вычитаний.

1.12. Дано вещественное число  $x$ . Не пользуясь никакими операциями, кроме умножения, получить

a)  $x^{21}$  за шесть операций                      b)  $x^3$  и  $x^{10}$  за четыре операции

c)  $x^5$  и  $x^{13}$  за пять операций                      d)  $x^2, x^5$  и  $x^{17}$  за шесть операций

e)  $x^4, x^{12}$  и  $x^{28}$  за шесть операций

1.13. Целой переменной  $k$  присвоить значение, равное третьей от конца цифре в записи целого положительного числа  $x$ .

1.14. Целой переменной  $k$  присвоить значение, равное сумме цифр в записи целого положительного трехзначного числа  $x$ .

1.15. Целой переменной  $k$  присвоить значение, равное первой цифре дробной части в записи вещественного положительного числа  $x$ .

1.16. Определить число, полученное выписыванием в обратном порядке цифр заданного целого трехзначного числа.

1.17. Идет  $n$ -ая секунда суток. Определить, сколько полных часов и полных минут прошло к этому моменту.

## ЛАБОРАТОРНАЯ РАБОТА № 2. «ПРЕОБРАЗОВАНИЕ ТИПОВ»

Цель работы: изучение команд преобразования типов.

Форма отчета: демонстрация выполненной работы преподавателю.

2.1. Пусть

char c; short s; int i; unsigned u; signed char sc;

float f; double d; long lng; unsigned short us; long double ld;

Определить тип выражений:

a) c - s/i

b) u \* 3 - 3.0 \* u - i

c) u - us \* i

d) (sc + d) \* ld

e) (5 \* lng - 'a') \* (s + u / 2)

f) (f + 3) / (2.5f - s \* 3.14)

2.2. Что будет напечатано в результате выполнения следующего фрагмента программы?

```
double d = 3.2, x; int i = 2, y;
```

```
x=(y=d/i) * 2; printf("x = %f; y = %d\n", x, y);
```

```
x=(y=d/i) * 2; printf("x = %d; y = %f\n", x, y);
```

```
y=(x=d/i) * 2; printf("x = %f; y = %d\n", x, y);
```

```
y=d * (x=2.5/d); printf("x = %f; y = %d\n", x, y);
```

```
x=d * (y = ((int)2.9 + 1.1)/d);
```

```
printf("x= %d y = %f\n", x, y);
```

2.3.

Модификаторы h и l (буква "эль") можно применять к спецификаторам d, i, o, u, x.

Модификатор h используется для вывода значения типа short int

Модификатор l позволяет выводить значения типа long int.

Модификатор l перед спецификаторами e, f используется для вывода значения типа long double.

Примеры

%u	целое число типа unsigned int;
%hd	целое число типа short со знаком в десятичной с.с;
%hu	целое число типа unsigned short;
%hx	целое число типа short со знаком в шестнадцатеричной с.с;
%ld	целое число типа long int со знаком в десятичной с.с;
%lu	целое число типа unsigned long int;
%lx	целое число типа long int со знаком в шестнадцатеричной с.с;
%lf	вещественный формат двойной точности (тип double);
%e	вещественный формат в экспоненциальной форме (числа с плавающей точкой типа float в экспоненциальной форме);

Что будет напечатано в результате выполнения следующего фрагмента программы?

double d; float f; long lng; int i; short s;

- a) `s = i = lng = f = d = 100/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- b) `d = f = lng = i = s =100/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n", s,i,lng,f,d);`
- c) `s = i = lng = f = d = 1000000/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- d) `d = f = lng = i = s =1000000/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- e) `lng = s = f = i = d =100/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- f) `f = s = d = lng = i = (double)100/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- g) `s = i = lng = f = d = 100/(double)3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- h) `f = s = d = lng = i = (double)100/3;`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`
- i) `i = s = lng = d = f = (double)(100/3);`  
`printf("s = %hd i = %d lng = %ld f = %f d = %f\n",s,i,lng,f,d);`

## ЛАБОРАТОРНАЯ РАБОТА № 3. «ОПЕРАТОРЫ СРАВНЕНИЯ. ЛОГИЧЕСКИЕ ОПЕРАТОРЫ»

Цель работы: получение навыков работы с логическими и операторами сравнения.

Форма отчета: демонстрация выполненной работы преподавателю.

3.1. Для верно записанных выражений вычислить их значения, определить тип результата (операции  $+$   $-$   $*$   $/$   $\%$   $++$  операции отношения, операции присваивания):

```
int d,i=1, j= 2, k=-7, m=0;
char c='w';
```

- |                     |                     |                        |
|---------------------|---------------------|------------------------|
| a) $d='a'+1<c;$     | b) $m=-i-5*j>=k+1;$ | c) $i+j++ +k == -2*j;$ |
| d) $m=3 < j < 5;$   | e) $m=3 == j < 5;$  | f) $m == c = 'w';$     |
| g) $m = c !=87;$    | h) $m=c!=87;$       | i) $m=!c+87;$          |
| j) $m=!c=87;$       | k)! $m ==c+87;$     | l) $m !=c+87;$         |
| m) $k == j-9 == i;$ | n) $i+j !=k;$       | o) $c+i < c-'x'+10;$   |
| p) $i-k=='0'+9< 10$ |                     |                        |

3.2. В логике справедливы утверждения: a)  $\text{not} (\text{not } x) = x;$  b)  $x \text{ and true} = x$

Верны ли соответствующие утверждения для операций  $!$  и  $\&\&$  в Си? Ответ обосновать.

3.3. При любом вещественном  $y > 0$  математически верно неравенство  $x < x+y$ . Верно ли подобное утверждение для выражения на Си?

3.4. Написать эквивалентное выражение, не содержащее операции  $!$

- |                            |                                    |                       |
|----------------------------|------------------------------------|-----------------------|
| a) $!(a>b);$               | b) $!(2*a == b+4);$                | c) $!(a<b \&\& c<d);$ |
| d) $!(a<2 \parallel a>5);$ | e) $!(a<1 \parallel b<2 \&\& c<3)$ |                       |

3.5. Выражения, соединенные операциями  $\&\&$  и  $\parallel$ , по правилам Си вычисляются слева направо; вычисления прекращаются, как только становится известна истинность или ложность результата. В других языках программирования, например в Паскале, вычисляются все части выражения в любом случае. Приведите «за» и «против» каждого из этих решений.

3.6. Допустимо ли в Си? Если "да" - опишите семантику этих действий; если "нет" - объясните почему.



## Технология программирования

```
a)int a, b, m, n, z;
m = n = 5;
z = a = b = 0;
z--, (a = b) = z + (m != n);
printf("%d %d %d %d
%d\n",a,b,m,n,z);
```

```
b)double x = 1.9; int a;
double b = 3.7;
a = b +=(1 && 2 || 3)!= (int)x;
printf ("%f %d %f\n", x, a,b);
```

```
c)int i, x, y;
x = 5; y = 10; i = 15;
x = ( y = 0, i = 1);
printf("%d %d %d\n", i, x ,y);
(x = y == 0) , i=1;
printf("%d %d %d\n", i, x, y);
```

```
d)int x, y;
x = 5;
y = x && ++ x;
printf("%d %d\n", x, y);
```

```
e)int x = 2, y, z;
x *= 3+2; x *= y = z = 4;
printf("%d %d %d\n", x, y, z);
x = y == z; x == ( y = z );
printf("%d %d %d\n", x, y, z);
```

```
f)int x = 2, y = 1, z = 0;
y = x && y||z;
x = x||!y&& z;
z = x / ++x;
printf("%d %d %d\n",x,y,z);
```

```
g)int x, y, z, i;
x = y = z = 1;
i = ++x || ++y && ++z;
printf("%d%d%d%d\n", x,y,z,i);
i = x++ <= --y || ++z >= i;
printf("%d%d%d%d\n", x,y,z,i);
```

3.7. Напечатать большее из а и b чисел, используя условное выражение " ? "

3.8. Допустимо ли в Си? Если "да" - опишите семантику этих действий; если "нет" - объясните почему.

```
int x, y, z;
x = y = z = 1;
x += y += z;
printf("%d\n",x<y?y++ : x++);
printf("%d\n",z+=x<y? ++x:y--);
printf("%d %d %d\n", x, y, z);
printf("%d\n", z>=y && y>=x);
```

## ЛАБОРАТОРНАЯ РАБОТА № 4. «ПОБИТОВЫЕ ОПЕРАЦИИ»

Цель работы: изучение принципа работы побитовых операций.

Форма отчета: демонстрация выполненной работы преподавателю.

4.1. Допустимо ли в Си? Если "да" - опишите семантику этих действий; если "нет" - объясните почему.

```
a) int x = 03, y = 02, z = 01;    b) int i;
printf("%d\n", x | y & -z);      i = (1|2)%(1|2);
printf("%d\n", x ^ y & -z);      printf("i = %d\n", i);
printf("%d\n", x & y && z);
printf("%d\n", x << 3);
```

```
c) int i = 1; i = i << i | i; printf("
i = %d\n", i);
```

4.2. «Упаковать» четыре символа в беззнаковое целое. Длина беззнакового целого равна 4.

4.3. «Распаковать» беззнаковое целое число в четыре символа. Длина беззнакового целого равна 4.

4.4. Заменить в целочисленной переменной  $x$   $n$  бит, начиная с позиции  $p$ ,  $n$  старшими инвертированными битами целочисленной переменной  $y$ .

4.5. Циклически сдвинуть значение целочисленной величины на  $n$  позиций вправо.

4.6. Циклически сдвинуть значение целочисленной величины на  $n$  позиций влево.

## ЛАБОРАТОРНАЯ РАБОТА № 5. «УСЛОВНЫЙ ОПЕРАТОР И ОПЕРАТОР SWITCH»

Цель работы: изучение принципа работы условного оператора и оператора Switch.

Форма отчета: демонстрация выполненной работы преподавателю.

5.1. Перечислить все ситуации, когда в программах на Си используется составной оператор.

5.2. Эквивалентны ли следующие фрагменты программы:

- 1) if (e1) if (e2) S1; else S2;
- 2) if (e1) {if (e2) S1; else S2;}
- 3) if (e1) {if (e2) S1;} else S2;
- 4) if (e1) if (e2) S1; else; else S2;
- 5) if (e1) if (e2) S1; else S2; else;

Замечание: здесь e1 и e2 - выражения допустимого в этом случае типа; S1 и S2 - произвольные операторы.

5.3. Определить какие ошибки встречаются в приведенных ниже фрагментах программы, и какие из фрагментов без ошибок эквивалентны между собой:

a) `x<5 y>=-10`  
`printf("%d",x+y);`

b) `x=45;y=-10;`  
`if (x<5)`  
`if (y>=-10)`  
`printf("%d",x+y);`

c) `x=45;y=-10;`  
`if (x<5,y>=-10)`  
`printf("%d",x+y);`

d) `x=45;y=-10;`  
`if (x<5&& y>=-10)`  
`printf("%d",x+y);`

5.4. Какой из фрагментов программы соответствует решению следующей задачи: если X положительное (не равное нулю) четное целое число, то присвоить X его квадрат, если не четное, то остаток от деления на три.

a) `if !(x<=0)`  
`if (x%2==0) x=x*x;`  
`else x%=3;`

b) `if (x>0)`  
`if (x%2!=0) x=x*x;`  
`else x%=3;`

c) `if !(x%2=0)`  
`if (x>0) x=x*x;`  
`else x%=3;`

d) `if !(x%2=0) {`  
`if (x>0) x=x*x;}`  
`else x%=3;`

5.5. Какие ошибки содержатся в приведенных ниже фрагментах программ:

a) int x;

....

```
switch (x) {
    case 5: x++; break;
    case x>0: x--; break;
    default: x+=66;
}
```

b) int x;

....

```
switch (x) {
    case 5: x+=7; break;
    case 6: case 11: --x;
    break;
}
```

c) int x;

....

```
switch (x) {
    case 5: x*=16;
    case 6: case 11: x-=23;
    break;
}
```

d) int x;

....

```
switch (5) {
    case 5: x+=7; break;
    case 6: case 11: --x;
    break;
}
```

5.6. Какие из задач можно решить с помощью оператора switch:

- Если остаток от деления  $X$  на 5 равен 2, то присвоить  $X$  значение 0, если остаток равен 3, то умножить  $X$  на  $-1$ .
- Если  $X$  равно 5, то присвоить  $Y$  квадрат  $X$ , если 7, то  $Y$  присвоить остаток от деления  $X$  на 2, в остальных случаях значение  $Y$  увеличить на 1.
- Если  $X$  больше 5, то увеличить его значение на 1, иначе вывести на печать  $Y$ .  $X$  и  $Y$  целые.

5.7. Определить, имеется ли среди заданных целых чисел  $A$ ,  $B$ ,  $C$  хотя бы одно чётное.

5.8. Даны три числа. Вывести на экран те из них, которые принадлежат отрезку  $[e, f]$ .

5.9. Определить, есть ли среди цифр заданного целого трёхзначного числа одинаковые.

5.10. Выбрать наибольшее из трёх заданных чисел.

5.11. Определить номер квадранта, в котором находится точка с координатами  $(x, y)$ .

5.12. Написать программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Увы – рабочий день!», «Ура! Суббота!», «Ура! Воскресенье!».

5.13. Написать программу, которая после ввода с клавиатуры числа (в диапазоне от 1 до 999), обозначает денежную единицу, дописывая слово «рубль» в правильной форме. Например, 12 рублей, 21 рубль и т. д.

5.14. Написать программу, вычисляющую стоимость междугородного разговора в соответствии с таблицей:

Город	Код	Цена руб/мин
Владивосток	432	9,20
Москва	095	4,10
Краснодар	861	2,05
Волгоград	844	2,50

Входными данными должны являться код города и число минут. На выходе вывести сообщение о стоимости минуты и сумме за разговор.

5.15. Проверить, поместится ли на диске компьютера музыкальная композиция, которая длится  $m$  минут и  $n$  секунд, если свободное дисковое пространство 6 мегабайт, а для записи одной секунды звука необходимо 16 килобайт.

5.16. Для нормального разведения золотых рыбок необходимо, чтобы на каждую рыбку в аквариуме приходилось не менее 3-х литров воды. По известным объему аквариума и количеству рыбок, в нем содержащихся, определить, является ли аквариум "перенаселенным" или нет, и указать количество рыбок, которых в случае перенаселенности необходимо поместить в другой аквариум.



```

while(y < 10 ) {x = y ++; z = ++y;}
printf(" x = %d y = %d z = %d\n", x, y, z);
d) for(y=1; y < 10; y++ ) x = y;
printf(" x = %d y = %d\n", x, y);
e) for(y = 1; (x = y ) < 10; y++ );
printf(" x = %d y = %d\n", x, y);
f) for(x = 0, y = 1000; y > 1; x++, y /= 10 )
printf(" x = %d y = %d\n", x, y);
    
```

6.8. Верно ли утверждение: «действие оператора continue; в приведенных ниже примерах эквивалентно действию оператора goto next;».

- while(E ) {S; ... continue; ... S; next: ;}
- do{S; ... continue; ... S; } while(E ); next: ; ...
- for(E1; E2; E3) {S; ... continue; ... S; next: ;}
- while(E) {S;...for(E1;E2;E3) {S; ... continue;...S;}...S; next:; }
- while(E) {S;...for(E1;E2;E3) {S;...continue;...S; next:; }... S;}

6.9. Верно ли утверждение: « действие оператора break; в приведенных ниже примерах эквивалентно действию оператора goto next; ».

- while(E) {S; ... break; ... S; next: ; }
- while(E) {S; ... break; ... S; } next: ; ...
- do {S;...break;...S;} while(E);next ; ...
- for(E1; E2; E3) {S; ...break;...S; next:;}
- while(E) {S;...for(E1;E2;E3) {S;...break;...S;} next:;...S;}
- while(E) {S;...for(E1;E2;E3) {S;...break;...S;}...S;next:;}

6.10. Подсчитать количество натуральных чисел  $n$  ( $11 \leq n \leq 999$ ), в записи которых есть

- две одинаковые цифры;
- только две одинаковые цифры.

6.11. Подсчитать количество натуральных чисел  $n$  ( $102 \leq n \leq 987$ ), в которых все три цифры различны.

6.12. Подсчитать количество натуральных чисел  $n$  ( $11 \leq n \leq 999$ ), являющихся палиндромами, и распечатать их.

6.13. Подсчитать количество цифр в десятичной записи целого неотрицательного числа  $n$ .

6.14. Определить, верно ли, что куб суммы цифр натурального числа  $n$  равен  $n^2$ .

6. Определить, является ли натуральное число  $n$  степенью числа 3.

6.15. Для данного вещественного числа  $a$  найти наименьшее натуральное число  $n$  такое, что сумма  $1 + (1/2) + (1/3) + \dots + (1/n)$  будет больше числа  $a$ .

6.16. Дано натуральное число  $n$ . Найти значение числа, полученного следующим образом: из записи числа  $n$  выбросить цифры 0 и 5, оставив прежним порядок остальных цифр.

6.17. Дано натуральное число  $n$ . Получить все его натуральные делители.

6.18. Дано натуральное число  $n$ . Получить все такие натуральные  $q$ , что  $n$  делится на  $q^2$  и не делится на  $q^3$ .

6.19. Дано целое число  $m > 1$ . Получить наибольшее целое  $k$ , при котором  $4k < m$ .

6.19. Распечатать первые  $n$  ( $n$  - задано) простых чисел ( $p$  - простое число, если  $p \geq 2$  и делится только на 1 и на себя).

6.20. Распечатать первые  $n$  ( $n$  - задано) чисел Фибоначчи:  
 $f_0 = 1; f_1 = 1; f_{k+1} = f_{k-1} + f_k; k = 1, 2, 3, \dots$

6.21. Вычислить значение  $\cdot i!$  для  $i$ , изменяющихся от 1 до  $n$ . Воспользоваться соотношением  $\cdot i! = 1 + 1 \cdot 2 + 1 \cdot 2 \cdot 3 + \dots + 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = 1 + 2 \cdot (1 + 3 \cdot (1 + \dots + n \cdot (1) \dots))$ .

6.22. Вычислить квадратные корни вещественных чисел  $x = 2.0, 3.0, \dots, 100.0$ . Распечатать значения  $x, \sqrt{x}$ . Количество итераций, необходимых для вычисления корня, определяется точностью  $\text{eps} > 0$  ( $\text{eps}$  - задано).

Для  $a > 0$  величина  $\cdot a$  вычисляется следующим образом:

$a_0 = 1; a_{i+1} = 0.5 \cdot (a_i + a/a_i) \quad i = 0, 1, 2, \dots$

Считать, что требуемая точность достигнута, если  $|a_i - a_{i+1}| < \text{eps}$ .

6.23. Для данного вещественного числа  $x$  и натурального  $n$  вычислить:

a)  $\sin x + \sin 2x + \dots + \sin nx$

b)  $\sin x + \sin x^2 + \dots + \sin x^n$

c)  $\sin x + \sin(\sin x) + \dots + \sin(\sin(\dots \sin(\sin x) \dots))$

6.24. Вычислить  $1 - 1/2 + 1/3 - 1/4 + \dots + 1/9999 - 1/10000$  следующими способами:

a) последовательно слева направо;

b) последовательно справа налево;

c) последовательно слева направо вычисляются  $1 + 1/3 + 1/5 + \dots + 1/9999$  и  $1/2 + 1/4 + \dots + 1/10000$ , затем второе значение вычитается из первого;

d) последовательно справа налево вычисляются  $1 + 1/3 + 1/5 + \dots + 1/9999$  и  $1/2 + 1/4 + \dots + 1/10000$ , затем второе значение вычитается из первого.

Сравнить и объяснить полученные результаты.

6.25. Натуральное число называется совершенным, если оно равно сумме всех своих делителей, за исключением самого себя. Дано натуральное число  $n$ . Получить все совершенные числа, меньшие



п.

6.26. Если  $p$  и  $q$  - простые числа и  $q = p+2$ , то они называются простыми сдвоенными числами или "близнецами" (twin primes). Например, 3 и 5 - такие простые числа. Распечатать все простые сдвоенные числа, меньшие  $N$ .

## ЛАБОРАТОРНАЯ РАБОТА № 7. «ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ ЯЗЫКА СИ»

Цель работы: изучение символьного типа данных.

Форма отчета: демонстрация выполненной работы преподавателю.

7.1. В этом задании необходимо ввести во входной поток последовательность символов, заканчивающуюся точкой. Каждое задание оформить в виде отдельной функции без параметров, возвращающей соответствующее заданию значение, с прототипом

**<тип возвращаемого значения> имя функции(void);**

Ввод описанной выше последовательности символов должен производиться в теле каждой создаваемой функции, предусмотреть вывод на экран сообщения типа "ВВЕДИТЕ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ". Таким образом, программа по окончании ее выполнения должна содержать кроме `main()` созданные функции.

а) определить, сколько раз в этой последовательности встречается символ 'a';

б) определить, сколько символов 'e' предшествует первому вхождению символа 'u' (либо сколько всего символов 'e' в этой последовательности, если она не содержит символа 'u');

в) выяснить, есть ли в данной последовательности хотя бы одна пара символов-соседей 'n' и 'o', т.е. образующих сочетание 'n' 'o' либо 'o' 'n';

г) выяснить, чередуются ли в данной последовательности символы '+' и '-', и сколько раз каждый из этих символов входит в эту последовательность;

д) выяснить, сколько раз в данную последовательность входит группа подряд идущих символов, образующих слово C++;

е) выяснить, есть ли среди символов этой последовательности символы, образующие слово char;

ж) выяснить, есть ли в данной последовательности фрагмент из подряд идущих символов, образующий начало латинского алфавита (строчные буквы), и какова его длина. Если таких фрагментов несколько, найти длину наибольшего из них. Если такого фрагмента нет, то считать длину равной нулю;

7.2. Вывести в выходной поток последовательность символов, измененную следующим образом:

## Технология программирования

- a) заменить все символы '?' на '!';
  - b) удалить все символы '-' и удвоить все символы '&';
  - c) удалить все символы, не являющиеся строчными латинскими буквами;
  - d) заменить все прописные латинские буквы строчными (другие символы копировать в выходной поток без изменения);
  - e) заменить все строчные латинские буквы прописными (другие символы копировать в выходной поток без изменения);
  - f) каждую группу рядом стоящих символов '+' заменить одним таким символом;
  - g) каждую группу из  $n$  рядом стоящих символов '\*' заменить группой из  $n/2$  рядом стоящих символов '+' ( $n \geq 2$ ); одиночные '\*' копировать в выходной поток без изменения;
  - h) удалить из каждой группы подряд идущих цифр все начальные незначащие нули (если группа состоит только из нулей, то заменить эту группу одним нулем);
  - i) удалить все комбинации символов the;
  - j) оставить только те группы цифр, которые составлены из подряд идущих цифр с возрастающими значениями; все остальные цифры и группы цифр удалить ( другие символы копировать в выходной поток без изменения);
  - k) заменить все комбинации символов child комбинациями символов children;
  - l) удалить группы символов, расположенные между фигурными скобками { и }. Скобки тоже должны быть удалены. Предполагается, что скобки сбалансированы, и внутри каждой пары скобок других фигурных скобок нет.
- 7.3. Вывести в выходной поток последовательность символов, измененную следующим образом:
- a) удалить из каждой группы подряд идущих цифр, в которой более двух цифр и которой предшествует точка, все цифры, начиная с третьей (например,  $a+12.3456-b-0.456789+1.3-45678$  преобразуется в  $a+12.34-b-0.45+1.3-45678$ );
  - b) удалить из каждой группы цифр, которой не предшествует точка, все начальные нули (кроме последнего, если за ним идет точка либо в этой группе нет других цифр, кроме нулей; например,  $a-000123+bc+0000.0008-0000+0001.07$  преобразуется в  $a-123+bc+0.0008-0+1.07$ ).

## ЛАБОРАТОРНАЯ РАБОТА № 8. «ОДНОМЕРНЫЕ МАССИВЫ»

Цель работы: получение навыков с одномерными массивами. Форма отчета: демонстрация выполненной работы преподавателю.

8.1. Числовые массивы. В функции `main( )` определить целочисленные (или вещественные) массивы `a` из 10 элементов и `b` из 20 элементов. Поочередно добавляя в `main( )` указанные ниже функции, обработать оба массива с их помощью. Вывести сообщения, соответствующие результатам обработки. При описании аргумента-массива использовать синтаксис описания формального параметра с указателем и с квадратными скобками.

a) Определить упорядочены ли строго по возрастанию элементы массива из  $n$  элементов.

b) Определить индекс первого элемента массива, значение которого равно заданному числу  $x$ . Если такого элемента в массиве нет, то считать номер равным  $-1$ .

c) Определить для массива  $x$  размерности  $n$  значение  $x_0 + x_0 * x_1 + x_0 * x_1 * x_2 + \dots + x_0 * x_1 * x_2 * \dots * x_m$ , где  $m$  - индекс первого отрицательного элемента этого массива либо число 0, если такого элемента в массиве нет.

d) Определить значение  $\max(x_0 + x_{n-1}; x_1 + x_{n-2}; x_2 + x_{n-3}; \dots x_{(n-1)/2} + x_{n/2})$ .

e) Определить значение  $\min(x_0 * x_1, x_1 * x_2, x_2 * x_3, \dots, x_{n-3} * x_{n-2}, x_{n-2} * x_{n-1})$ ,

f) Определить значение  $u_0 * z_0 + u_1 * z_1 + \dots + u_k * z_k$ , где  $u_i$  - отрицательные элементы массива  $x$ , взятые в порядке их следования;  $z_i$  - положительные элементы этого массива, взятые в обратном порядке;  $k = \min(p, q)$ , где  $p$  - количество положительных элементов массива  $x$ ,  $q$  - количество отрицательных элементов этого массива.

8.2. Массивы символов. Выполнить задания из раздела 7 с использованием строк (длина вводимой последовательности символов не превысит 100 элементов.)

## ЛАБОРАТОРНАЯ РАБОТА № 9. «СТРОКИ»

Цель работы: получение навыков с одномерными массивами.

Форма отчета: демонстрация выполненной работы преподавателю.

Цель работы: получение навыков работы со строками.

Форма отчета: демонстрация выполненной работы преподавателю.

9.1. Написать программу, в которой:

- описать массив символов `s` и инициализировать его строкой "1234567890";
- вычислить длину строки и динамически выделить память для копии строки;
- выполнить копирование. Вывести копию строки на экран (с поясняющим текстом, что напечатана копия);
- с помощью функции `strncpy` заменить в копии строки три символа, начиная со второго. Для замены ввести короткую строку с клавиатуры;
- вывести преобразованную строку на экран.

9.2. Написать программу, в которой:

- с клавиатуры вводятся четыре строки разной длины (не более 10 символов). Память для каждой из них выделяется динамически;
- необходимо объединить эти строки их в одну, память для которой выделяется в соответствии с потребностью динамически.

9.3. Написать программу, в которой:

- ввести четыре символьные строки;
- найти среди них "наибольшую". Промежуточные результаты выводить на экран в виде сообщения типа <строка-1> "больше"/"меньше" <строка-2>;
- отдельно напечатать "наибольшую" строку.

9.4. Написать программу, в которой:

- создать список фамилий в виде одной символьной строки. Например, "1. Иванов И.И. 2. Петров П.П. 3. Сидоров С.С. 4. Александров А.А.";
- в цикле в режиме диалога вводить фамилию и выводить сообщение, есть ли такой человек в списке. Признак конца диалога - пустая строка.

9.6. Написать программу, в которой:

- описать строку `s` (как в задании №1);
- заменить символы с третьего по пятый на "минус", а с шестого и до конца строки - на "плюс";

- вывести на экран исходную строку, промежуточный и окончательный результат.

## **ЛАБОРАТОРНАЯ РАБОТА № 10. «СТАНДАРТНАЯ БИБЛИОТЕКА СИ ДЛЯ РАБОТЫ С СИМВОЛЬНЫМИ СТРОКАМИ»**

Цель работы: получение навыков с одномерными массивами.  
Форма отчета: демонстрация выполненной работы преподавателю.  
Цель работы: получение навыков работы с функциями стандартной библиотеки.

Форма отчета: демонстрация выполненной работы преподавателю.

Задание 1. Написать программу, в которой

- описать массив символов `s` и инициализировать его строкой "1234567890".
- вычислить длину строки и динамически выделить память для копии строки.
- выполнить копирование. Вывести копию строки на экран (с поясняющим текстом, что напечатана копия).
- с помощью функции `strncpy` заменить в копии строки три символа, начиная со второго. Для замены вводим короткую строку с клавиатуры.

Задание 2. Написать программу, в которой

- с клавиатуры вводятся четыре строки разной длины (не более 10 символов). Память для каждой из них выделяется динамически.
- необходимо объединить эти строки их в одну, память для которой выделяется в соответствии с потребностью динамически.

Задание 3. Написать программу, в которой

- ввести четыре символьные строки;
- найти среди них "наибольшую". Промежуточные результаты выводить на экран в виде сообщения типа  
    <строка-1> "больше"/"меньше" <строка-2>.
- Отдельно напечатать "наибольшую" строку.

Задание 4. Написать программу, в которой

- создать список фамилий в виде одной символьной строки. Например, "1. Иванов И.И. 2. Петров П.П. 3. Сидоров С.С. 4. Александров А.А."
- в цикле в режиме диалога вводить фамилию и выводить сообщение, есть ли такой человек в списке. Признак конца диалога - пустая строка.

Задание 5. Допisać фрагмент программы, выполнить ее, пояснить

текст и результаты.

```
char *ptr;
ptr=strtok("Feb. 15, 2001", ". /");
printf("%s/n", ptr);
while(ptr){
    ptr=strtok(NULL, ". - /");
    if(ptr) printf("%s/n", ptr); //Зачем здесь нужна проверка ptr?
}
```

Задание 6. Написать программу, в которой описать строку s (как в задании №1). Заменить символы с третьего по пятый на "минус", а с шестого и до конца строки - на "плюс". Вывести на экран исходную строку, промежуточный и окончательный результат.

Задание 7. Написать свою функцию str\_dup(), которая работает аналогично библиотечной функции strdup().

## **СПИСОК ЛИТЕРАТУРЫ**

1. В. В. Подбельский, С. С. Фомин Курс программирования на языке Си: учебник. – М.: ДМК Пресс, 2012. – 384 с.
2. Герберт Шилдт С. Полное руководство. – М. : Диалектика, 2017. – 704 с.