

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №7а

з дисципліни «Програмування»

Підготував:
ст. групи АП-11
Чаус В. О.

Прийняла:
Чайковський І. Б.,

Тема:

Арифметичні операції та вирази мови C.

Мета:

Дослідження принципів створення математичних виразів при складанні програм для виконання обчислень за допомогою різних операцій мови програмування C.

Теоретичні відомості:

Елементарною коміркою машинної пам'яті є біт. Біт – це елемент інформації, який може приймати значення 1 або 0. Фізично це означає наявність або відсутність електричного струму в певній ділянці електричного кола. Такий спосіб представлення елементу інформації пристосований для двійкової системи числення, яка використовується в ЕОМ. Група з восьми біт утворює байт. В одному байті можна записати беззнакове ціле число від 0 до 255 (256 - восьмий степінь числа 2) або знакове від 0 до 127. Звичайно одного байту недостатньо для запису більш складних даних, тому з двох (або чотирьох) байт утворюється машинне слово - вектор бітів, який розглядається апаратною частиною ЕОМ як єдине ціле. Число бітів у слові називається довжиною слова, залежить від апаратної реалізації комп'ютера і, як правило, буває довжиною 16 або 32 біти. Пам'ять обчислювальної машини поділяється логічно на слова. Слово має довжину, достатню для розміщення в ньому команди або цілого числа.

Завдання:

1. Здійснити виконання програми VALUES.C:

```
#include <stdio.h>
#include <conio.h>
int main()
{
printf("Числа типу int займають %d байт.\n",sizeof(int));
printf("Числа типу char займають %d байт.\n",sizeof(char));
printf("Числа типу float займають %d байт.\n",sizeof(float));
printf("Числа типу double займають %d байт.\n",sizeof(double));
getch();
}
```

2. Створити і виконати програми дослідження властивостей арифметичних операцій із різними типами величин.

```
// Префіксний та постфіксний
// інкремент ++ і декремент --
#include <stdio.h>
#include <conio.h>
int main()
{ int n = 1;
```

```

printf("n=%d \n",n);
// n++;
printf("prefix: ++n=%d\n",++n);
printf("postfix: n++=%d\n",n++);
printf("after-postfix: n=%d\n",n);
// n--;
printf("prefix: --n=%d\n",--n);
printf("postfix: n--=%d\n",n--);
printf("after-postfix: n=%d\n",n);
}

```

3. Виконати завдання згідно варіанта і пояснити результат при $n=1$, $m=1$.
Зразок програми.

```

int main()
{
int n=1,m=1,res1,res2;
res1=n+(m---);
printf("res1=%d\n",res1);
res2=m--+n;
printf("res2=%d",res2);
return 0;
}

```

9	1) ++n*++m
	2) m++-n

4. Виконати приклади і пояснити результати

```

#include <stdio.h>
#include <conio.h>
int main()
{ int a, b=3;
float c;
c = b%2 + (a = ++b/2) + 1.1;
printf("a=%d,c=%4.1f\n",a,c);
}

#include <stdio.h>
int main()
{
int x=2,z;
float y = 2.1;
z = x++*y + y/x*3;
printf("x=%d z=%d\n",x,z);
}

#include <stdio.h>
int main()
{


```

```
float x = 1.1, y = 0, z;  
int a;  
z = (a=x++)*y + 3*x;  
printf("z=%4.1f\n",z); }
```

Виконання роботи:

1.

```
1 #include <stdio.h>  
2 #include <conio.h>  
3  
4 int main() {  
5     printf("Числа типу int займають %zu байт.\n", sizeof(int));  
6     printf("Числа типу char займають %zu байт.\n", sizeof(char));  
7     printf("Числа типу float займають %zu байт.\n", sizeof(float));  
8     printf("Числа типу double займають %zu байт.\n", sizeof(double));  
9     getch();  
10    return 0;  
11 }  
12
```



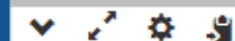
The screenshot shows the output of the program in a terminal window. The output is as follows:

```
Числа типу int займають 4 байт.  
Числа типу char займають 1 байт.  
Числа типу float займають 4 байт.  
Числа типу double займають 8 байт.
```

Виводимо на екран повідомлення, скільки байтів має певний тип, і так для всіх типів

2.

```
1  #include <stdio.h>
2  #include <conio.h>
3
4  int main() {
5      int n = 1;
6
7      printf("n=%d \n", n);
8
9      printf("prefix: ++n=%d\n", ++n);
10
11     printf("postfix: n++=%d\n", n++);
12
13     printf("after-postfix: n=%d\n", n);
14
15     printf("prefix: --n=%d\n", --n);
16
17     printf("postfix: n--=%d\n", n--);
18
19     printf("after-postfix: n=%d\n", n);
20
21     getch();
22     return 0;
23 }
24
```



```
n=1
prefix: ++n=2
postfix: n++=2
after-postfix: n=3
prefix: --n=2
postfix: n--=2
after-postfix: n=1
```

Задаємо змінну та значення, далі додаємо одразу 1 і виводимо, потім додаємо 1 після виведення на екран, далі віднімаємо 1 і виводимо, потім виводимо змінну і віднімаємо 1. В кінці виводимо цю змінну

3.

```
1  #include <stdio.h>
2
3  int main() {
4      int n = 1, m = 1, res1, res2;
5
6      res1 = ++n * ++m;
7      printf("res1=%d\n", res1);
8
9      res2 = m++ - n;
10     printf("res2=%d", res2);
11
12     return 0;
13 }
14
```

```
res1=4
res2=0
```

Задаємо чотири змінні, для двох вказуємо значення. Далі в третю змінну заносимо результат: додаємо 1 до змінної n тепер її множимо до m , до якої вже додали 1

В четвертій змінній ми додаємо пізніше 1 до m і віднімаємо n

Все це виводимо на екран

4.

```
1  #include <stdio.h>
2  #include <conio.h>
3
4  int main() {
5      int a, b = 3;
6      float c;
7
8      c = b % 2 + (a = ++b / 2) + 1.1;
9
10     printf("a=%d,c=%4.1f\n", a, c);
11
12     getch();
13     return 0;
14 }
15
```



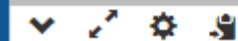
a=2,c= 4.1

Задаємо змінні, та їх значення.

В одну змінну заносимо вираз, в якому додатково задаємо значення для змінній a, і далі по виразу

В результаті виводимо змінні a,c

```
1  #include <stdio.h>
2
3  int main() {
4      int x = 2, z;
5      float y = 2.1;
6
7      z = x++ * y + y / x * 3;
8
9      printf("x=%d z=%d\n", x, z);
10
11     return 0;
12 }
13
```



x=3 z=6

Задаємо дві змінні та їх значення, в третю задаємо вираз
В результаті виводимо x,z


```
1  #include <stdio.h>
2
3  int main() {
4      float x = 1.1, y = 0, z;
5      int a;
6
7      z = (a = x++) * y + 3 * x;
8
9      printf("z=%4.1f\n", z);
10
11     return 0;
12 }
13
```

z= 6.3

Задаємо дві змінні та їх значення, третю, п'яту просто ініціалізуємо, в четверту задаємо вираз в якому задаємо значення п'ятої змінні, і далі йдемо по виразу. В результаті виводимо z.