

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  
«Програмування, частина 2»

Підготував:  
ст. гр. ІХ-11  
Диркавець Максим  
Перевірив:  
Асистент каф РЕПС  
Чайковський  
І.Б.

**Тема:** Арифметичні операції та вирази мови C.

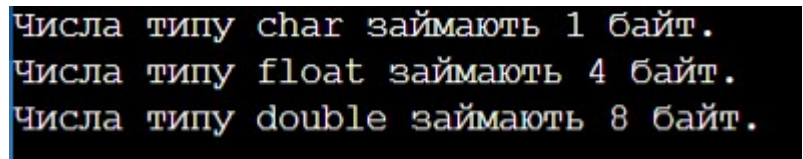
**Мета роботи:** Дослідження принципів створення математичних виразів при складанні програм для виконання обчислень за допомогою різних операцій мови програмування C.

## Хід роботи

### Завдання 1

```
#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();
}
```



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

### Завдання 2

```
#include <stdio.h>

int main() {
    int n = 1;
    printf("n=%d \n", n);
    printf("prefix: ++n=%d\n", ++n);
    printf("postfix: n++=%d\n", n++);
    printf("after-postfix: n=%d\n", n);
}
```

```

    printf("prefix: --n=%d\n", --n);
    printf("postfix: n--=%d\n", n--);
    printf("after-postfix: n=%d\n", n);
    return 0;
}

```

```

prefix: --n=2
postfix: n--=2
after-postfix: n=1

```

### Завдання 3

3	1) n---m 2) m--*n
---	----------------------

```

#include <stdio.h>

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;
}

```

```

res1 = 2
res2 = 1

```

### Завдання 4

1 КОД

```

#include <stdio.h>

```

```
#include <conio.h>

int main()
{
    int a, b = 3;
    float c;
    c = b % 2 * 2 + (a = b / 2) + 1.1;
    printf("a=%d, c=%4.1f\n", a, c);
    return 0;
}
```

```
a=1, c= 4.1
```

Пояснення:

**b = 3**

**b % 2** обчислює залишок від ділення 3 на 2, що дає 1.

**1 \* 2** дає 2.

**(a = b / 2)** виконує ділення **3 / 2**, що дає 1 (в цілому числі), і присвоює це значення **a**.

Потім ми маємо **2 + 1 + 1.1**, що дає 4.1.

**Результат a = 1 та c = 4.1.**

2 КОД

```
#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);
    return 0;
}
```

```
}
```

```
x=3 z=6
```

Пояснення:

$x = 2$

$y = 2.1$

$x++$  використовує поточне значення  $x$  (яке є 2), а потім збільшує  $x$  на 1. Тобто,  $x$  стає 3 після цієї операції, але для обчислення використовується 2.

Результат  $2 * 2.1 = 4.2$ .

Потім обчислюється  $y / x * 3$ , де  $y = 2.1$  і нове значення  $x = 3$ . Отже,  $2.1 / 3 * 3 = 2.1$ .

Сума  $4.2 + 2.1$  дає  $6.3$ .

Змінна  $z$  є цілим числом, тому вона буде 6 після відкидання дробової частини.

3 КОД

```
#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);
```

```
    return 0;
```

```
}
```

```
z= 6.3
```

Пояснення:

$x = 1.1$

$x++$  використовує поточне значення  $x$  (1.1), а потім збільшує  $x$  на 1. Тобто,  $x$  стає 2.1 після цієї операції.

$a = 1$  (оскільки  $a$  отримує цілу частину  $x$  до інкременту).

Результат  $(a = 1) * y = 1 * 0 = 0$ .

Потім обчислюється  $3 * x$ , де  $x = 2.1$ , тобто  $3 * 2.1 = 6.3$ .

Сума  $0 + 6.3 = 6.3$ .

**Висновок:** На лабораторній даній роботі ми дослідили принципи створення та використання арифметичних виразів у мові програмування C. Розглянули базові арифметичні операції, а також постфіксні оператори інкременту та декременту. Виконані приклади показали важливість порядку операцій та правильного використання типів даних. Це підкреслює необхідність точного розуміння арифметичних операцій для ефективного програмування на мові C.