

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



З В І Т
до лабораторної роботи №7а

на тему:
«Арифметичні операції та вирази мови C»

Спеціальність 6.172.00
«Телекомунікації та радіотехніка»

*Виконав: ст. гр. АП-11
Чума Трас.
Прийняла:
Гордійчук-Бублівська О. В.*

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

Завдання:

1.

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



The screenshot shows the output of the C program in a terminal window. The output consists of four lines of text, each preceded by a vertical bar character '|'. The text indicates the memory size in bytes for the types int, char, float, and double. To the right of the output, there is a small diagram showing the memory layout with labels 'int' and 'long unsigned int' and a format specifier '%ld'.

Тип	Розмір (байт)
int	4
char	1
float	4
double	8

2.

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



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

3.1

```
main.c
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      return 0;
9  }
10
```

res1=0

Для $res1 = n++ - m$:

- 1) “n++” поверне поточне значення n (1), а потім збільшить його до 2
- 2) “m” дорівнює 1.

$$res1 = 1 - 1 = 0$$

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

res2=1

Для $res2 = m-- * n$:

- 1) m-- спочатку поверне поточне значення m (1), а потім зменшить його до 0.
- 2) “n” дорівнює 1.

$$res2 = 1 * 1 = 1$$

4.1

```
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     return 0;
13 }
```

▼ ↗ ⚙ 🖨

a=2, c= 4.1

4.2

```
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

4.3

```
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

Висновок:

Лабораторна робота №7а на тему "Арифметичні операції та вирази мови С" дозволила мені отримати глибше розуміння роботи арифметичних операцій у мові програмування С та їх використання в складних виразах. Отже, лабораторна робота №7а дозволила мені розширити знання про арифметичні операції та вирази мови С, що буде корисним у подальшому програмуванні.