

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Кафедра систем штучного інтелекту

**Звіт
до лабораторної роботи
на тему: «Знайомство з С.
Виконання програми простої структури.»
з дисципліни:
«Алгоритмізація та програмування»**

Виконав:
студент групи КН-110
Романьчук Максим
Викладач:
Гасько Р.Т.

Львів 2018 р.

Мета

Знайомство з середовищем програмування, створення, відлагодження й виконання простої програми, що містить ввід/вивід інформації й найпростіші обчислення.

Завдання для варіанту 24

Завдання 1:

Обчислити значення виразу при різних дійсних типах даних (float й double). Обчислення варто виконувати з використанням проміжних змінних. Порівняти й пояснити отримані результати.

Вираз:

$$\frac{(a-b)^4 - (a^4 - 4a^3b + 6a^2b^2)}{b^4 - 4ab^3}$$

При $a = 1000$, $b = 0,0001$.

Розв'язання:

Для виконання цього завдання я використовував по десять змінних типу float і double для перенесення виразу в код та використання функції піднесення до степеня (pow).

Код програми:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    /*Exercise 1: Calculation of the same formula with different types of variables.
    Part 1: float type calculation.*/
    float pt1, pt2, n;
    float b = 1000.0;
    float a = 0.0001;
    float ab4 = pow(a - b, 4); /*Піднесення частин виразу до степенів*/
    float a4 = pow(a, 4), a3 = pow(a, 3), a2 = pow(a, 2);
    float b4 = pow(b, 4), b3 = pow(b, 3), b2 = pow(b, 2);
    pt1 = (ab4 - (a4 - 4*a3*b + 6*a2*b2)); /*Обчислення чисельника*/
    pt2 = (b4 - 4*a*b3); /*Обчислення знаменника*/
    n = pt1 / pt2;
    printf("1)float=%f\n", n); /*Виведення результату*/
}

/*Part 2: double type calculation.*/
double p1, p2, m;
double b = 1000.0;
double a = 0.0001;
float ab4 = pow(a - b, 4);
double a4 = pow(a, 4), a3 = pow(a, 3), a2 = pow(a, 2);
double b4 = pow(b, 4), b3 = pow(b, 3), b2 = pow(b, 2);
p1 = (ab4 - (a4 - 4*a3*b + 6*a2*b2));
p2 = (b4 - 4*a*b3);
m = p1 / p2;
printf("2)double=%.51f\n", m);
}
```

Результат:

```
1)float=1.000000
2)double=1.000000002687941
```

Пояснення до результату:

Різниця в результатах зумовлена в точності чисел, якими обчислювався вираз. Адже тип float має вдвічі меншу кількість знаків після крапки, що обчислюються, аніж тип double.

Завдання 2:

Обчислити значення виразів. Пояснити отримані результати.

Вирази:

1) $n++*m$

2) $n++<m$

3) $m-->m$

Розв'язання:

Для виконання цього завдання я використовував функції printf та scanf, щоб програма отримувала значення від користувача в форматі int а у виразах 2 і 3 результат перетворився в true або false (залежно від результату порівняння).

Код програми:

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    /*Exercise 2: Operators.Decrements and Increments. Comparing operators.*/
    int m, n, r1, r2, r3, z;
    /*Step 1: Reading arguments*/
    printf("Please, write first number(n):");
    scanf("%i", &n);
    printf("Ok, write another one(m):");
    scanf("%i", &m);
    /*Step 2: Calculation*/
    r1 = n++*m;
    n--;
    r2 = n++<m;
    z = m;
    r3 = m-->z;
    printf("2.1) n++*m=%i\n", r1);
    /*Step 3: Showing results (2.2 and 2.3 are boolean expressions, so they can only be true(1) or false(0))*/
    if(r2 == 1)
    {
        if(r3 == 1)
        {
            printf("2.2) n++<m=true\n"); /*different combinations of possible results*/
            printf("2.3) m-->m=true\n");
        }
        else
        {
            printf("2.2) n++<m=true\n");
            printf("2.3) m-->m=false\n");
        }
    }
    else if(r3 == 1)
    {
        printf("2.2) n++<m=false\n");
        printf("2.3) m-->m=true\n");
    }
    else
    {
        printf("2.2) n++<m=false\n");
        printf("2.3) m-->m=false\n");
    }
}
```

Результат:

```
Please, write first number(n):1  
Ok, write another one(m):1  
2.1) n++*m=1  
2.2) n++<m=false  
2.3) m-->m=false
```

(як приклад використовувались одиниці)

Пояснення до результату:

Інкременти n в першому та другому виразах не змінюють результату, адже його дія виконується з наступного рядка, те ж з декрементом в третьому виразі. Сам третій вираз завжди невірний, адже жодне число не може бути більше за себе.