

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра систем штучного інтелекту



Лабораторна робота №1
З курсу “Алгоритмізація та програмування”

Виконав:
ст.гр. КН-110
Холод Ігор

Львів – 2018

1)Завдання 1.

Обчислити значення виразу при різних дійсних типах даних float і double.

$$((a + b)^3 - a^3)/(3ab^2 + b^3 + 3a^2b)$$

Код для даної програми виглядає так:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float a = 100, b = 0.0001, fr, fa, fb, fc, fd, fe;
7     double a1 = 100, b1 = 0.0001, dr, da, db, dc, dd, de;
8
9     fa = pow((a + b), 3);
10    fb = pow (a, 3);
11    fc = 3*a*pow(b, 2);
12    fd = pow(b, 3);
13    fe = 3*pow(a, 2)*b;
14    fr = (fa - fb)/(fc + fd + fe);
15
16    da = pow((a1 + b1), 3);
17    db = pow (a1, 3);
18    dc = 3*a1*pow(b1, 2);
19    dd = pow(b1, 3);
20    de = 3*pow(a1, 2)*b1;
21    dr = (da - db)/(dc + dd + de);
22
23    printf ("Float result is %1.13f\n", fr);
24    printf ("Double result is %1.13f\n", dr);
25
26    return 0;
27 }
```

Результат, який повертає ця програма:

```
Float result is 0.9999989867210
Double result is 1.00000000000461
```

Відмінності обчислень при використанні різних типів даних пояснюється тим, що float має вдвічі меншу точність (зберігає вдвічі меншу кількість знаків після коми, ніж double)

Завдання 2.

Визначити результат даних обчислень (значення m і n вводяться користувачем):

1) $n++ * m$

2) $n++ < m$

3) $m-- > m$

Код програми виглядає так:

```
1 #include "stdio.h"
2
3 int main()
4 {
5     int m, n, res1, res2, res3;
6
7     printf ("Input m = ");
8     scanf ("%i", &m);
9     printf ("Input n = ");
10    scanf ("%i", &n);
11
12    res1 = n++ * m;
13    printf("n++ * m (%i * %i) = %i.\n", n - 1, m, res1);
14    res2 = n++ < m;
15    if (res2 == 1) printf ("n++ < m (%i < %i) is a TRUE statement!\n", n - 1, m);
16    else if (res2 == 0) printf ("n++ < m (%i < %i) is a FALSE statement!\n", n - 1, m);
17    else printf ("Something went wrong!\n");
18    int temp = m;
19    res3 = m-- > --temp;
20    if (res3 == 1) printf ("m-- > m (%i > %i) is a TRUE statement!\n", m + 1, m);
21    else if (res3 == 0) printf ("m-- > m (%i > %i) is a FALSE statement!\n", m + 1, m);
22    else printf ("Something went wrong!\n");
23    return 0;
24 }
```

Спочатку користувач вводить значення m і n (у цьому випадку 6 і 7, відповідно).

Потім програма по чергово рахує значення виразів.

Якщо інкремент\декремент стоїть після змінної, то спочатку береться значення змінної, виконується задана операція, і тоді змінній присвоюється значення на 1 більше/менше. Якщо перед, то спочатку змінній присвоюється значення на 1 більше/менше, а тоді виконується операція.

1) $n++ * m$: Береться значення m , множиться на n , і після цих операцій змінній n присвоюється значення на 1 більше.

2) $n++ < m$: Береться нове значення n і порівнюється з m . Якщо нерівність невірна, змінній $res2$ присвоюється значення 0, а якщо вірна - то 1. Після операції змінній присвоюється значення на 1 більше.

3) $m-- > m$: Аналогічно порівнюються значення $m--$ і m . Вдруге m матиме на 1 менше значення, оскільки перед цим ми вже записали $m--$, отже ця вірність завжди буде вірною.

Результат:

```
Input m = 6
Input n = 7
n++ * m (7 * 6) = 42.
n++ < m (8 < 6) is a FALSE statement!
m-- > m (6 > 5) is a TRUE statement!
```

Висновок: змінну `double` краще використовувати, коли потрібна досить велика точність обчислень, чого `float` не завжди може забезпечити. Також між преінкрементом/предекрементом і постінкрементом/постдекрементом є суттєва різниця: в першому випадку операція відбувається перед використанням змінної, а в другому - після.