

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота**  
з дисципліни  
«Алгоритмізація та програмування»

Виконала:  
студентка КН-109  
Дипко Олександра  
Викладач:  
Варецький Я.

Львів – 2018 р.

### **Постановка 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.
2. Обчислити значення виразів. Пояснити отримані результати

## Варіант №6

### 1. Обчислити значення виразу при різних дійсних типах даних (float і double)

$$\frac{(a-b)^3 - (a^3 - 3ab^2)}{b^3 - 3a^2b} \quad \text{при } a=1000, b=0.0001$$

#### Double:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     double a,b,c;
7
8     printf ("enter a: \n");
9     scanf ("%lf", &a);
10    printf ("enter b: \n");
11    scanf ("%lf", &b);
12
13    c = ( pow(a-b, 3) - ( pow (a, 3) - 3*a*pow(b,2)))/( pow (b,3) - 3*pow(a,2)*b);
14
15    printf ( "Result: %f \n", c );
16
17    return 0;
18 }
```

#### Float:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     double a,b,c;
7
8     printf ("enter a: \n");
9     scanf ("%lf", &a);
10    printf ("enter b: \n");
11    scanf ("%lf", &b);
12
13    c = ( pow(a-b, 3) - ( pow (a, 3) - 3*a*pow(b,2)))/( pow (b,3) - 3*pow(a,2)*b);
14
15    printf ( "Result: %f \n", c );
16
17    return 0;
18 }
```

#### Result:

```
jharvard@appliance (~): cd proj
jharvard@appliance (~/.proj): make labfloat
make: `labfloat' is up to date.
jharvard@appliance (~/.proj): ./labfloat
enter a:
1000
enter b:
0.0001
Result: 1.220703
jharvard@appliance (~/.proj): ./labdouble
enter a:
1000
enter b:
0.0001
Result: 1.000000
jharvard@appliance (~/.proj):
```

Значення виразу  $\sim 1$ . Оскільки змінні типу float мають малу точність, то при обрахунках з великою кількістю знаків після крапки частина інформації втрачається, а отже й точність теж. Отже, точність double вдвічі перевищує точність float. Змінні з типом double мають удвічі більшу кількість знаків після крапки, а отже й кращу точність при обчисленні таких виразів.

## 2. Обчислити значення виразів. Пояснити отримані результати.

### №1

**m-++n**

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float m,n,c;
7
8     printf ("write m: ");
9     scanf("%f", &m);
10
11     printf ("write n: ");
12     scanf("%f", &n);
13
14     c = m - (++n);
15     printf ("The answer is %f \n", c);
16
17
18     return 0;
19 }
```

```
jharvard@appliance (~): cd proj
jharvard@appliance (~/proj): make 1
make: `1' is up to date.
jharvard@appliance (~/proj): ./1
write m: 5
write n: 5
The answer is -1.000000
jharvard@appliance (~/proj):
```

Спочатку (n) набуває значення на одиницю менше, тоді значення віднімаються. При m=5, n=5 результатом є -1.

## №2

**++m>--n**

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float m,n,c;
7
8     printf ("write m: ");
9     scanf("%f", &m);
10
11     printf ("write n: ");
12     scanf("%f", &n);
13
14     c = (++m) > (--n);
15     printf ("The answer is %f \n", c);
16
17
18     return 0;
19 }
```

```
cdjharvard@appliance (~): cd proj
jharvard@appliance (~/proj): make 2
make: `2' is up to date.
jharvard@appliance (~/proj): ./2
write m: 4
write n: 4
The answer is 1.000000
jharvard@appliance (~/proj):
```

```
jharvard@appliance (~): cd proj
jharvard@appliance (~/proj): make 2
make: `2' is up to date.
jharvard@appliance (~/proj): ./2
write m: 2
write n: 5
The answer is 0.000000
jharvard@appliance (~/proj):
```

При m=4, n=4 рівність виконується, отримаємо результат 1. При m=2, n=5 рівність не виконується, отримаємо 0.

**--n<++m**

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float m,n,c;
7
8     printf ("write m: ");
9     scanf ("%f", &m);
10
11     printf ("write n: ");
12     scanf ("%f", &n);
13
14     c = (--n) < (++m);
15     printf ("The answer is %f \n", c);
16
17     |
18     return 0;
19 }

```

```

jharvard@appliance (~): cd proj
jharvard@appliance (~/.proj): make 3
make: `3' is up to date.
jharvard@appliance (~/.proj): ./3
write m: 5
write n: 5
The answer is 1.000000

```

```

jharvard@appliance (~): cd proj
jharvard@appliance (~/.proj): make 3
make: `3' is up to date.
jharvard@appliance (~/.proj): ./3
write m: 2
write n: 5
The answer is 0.000000
jharvard@appliance (~/.proj): █

```

При m=2, n=5 рівність не виконується, отримаємо результат 0. При m=5, n=5 рівність виконується, отримаємо 1.