**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ**

**“ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота**

**з дисципліни**

**«Алгоритмізація та програмування»**

**Виконав:**

**студент групи КН-111**

**Пучак Віталій**

**Викладач:**

**Гасько Р.Т.**

**Львів – 2018 р.**

**Зміст звіту**

1. Постановка завдання.

2. Програма розв’язку завдання1.

3. Результати роботи програми для даних типу float.

4. Результати роботи програми для даних типу double.

5. Пояснення результатів.

6. Програма розв’язку завдання2.

7. Результати роботи програми.

8. Пояснення результатів.

**1. 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 результат

**2. Програма розв’язку завдання 1.**

#include<stdio.h>

#include<math.h>

int main(void)

{

float a=1000,b=0.0001,c;

c=(pow(a+b,2)-(pow(a,2)+(2\*a\*b)))/pow(b,2);

double ad=1000,bd=0.0001, cd;

cd=(pow(ad+bd,2)-(pow(ad,2)+(2\*ad\*bd)))/pow(bd,2);

printf("Float:%f\nDouble:%f\n",c,cd);

return 0;

}

**3. Результати роботи програми для даних типу float**.

4414065.5

**4. Результати роботи програми для даних типу double.**

1.001172

**5. Пояснення результатів.**

При а=1000 і b=0.0001 рівняння ≈ 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.

З іншого боку, змінні з типом double мають удвічі більшу кількість

знаків після крапки, а отже й кращу точність при обчисленні таких

виразів.

Результат обчислення з типом float більший від результату з double

через піднесення змінної b до степення, при якому вона ставала

надто малою для збереження у float.

**6. Програма розв’язку завдання2.**

#include<stdio.h>

int main(void)

{

int n,m;

printf("Enter n:");

scanf("%d",n);

printf("Enter m:");

scanf("%d",m);

printf("1)%d\n",n+++m);

if(m-->n)

printf("2)True\n");

else

printf("2)False\n");

if(n-->m)

printf("3)True\n");

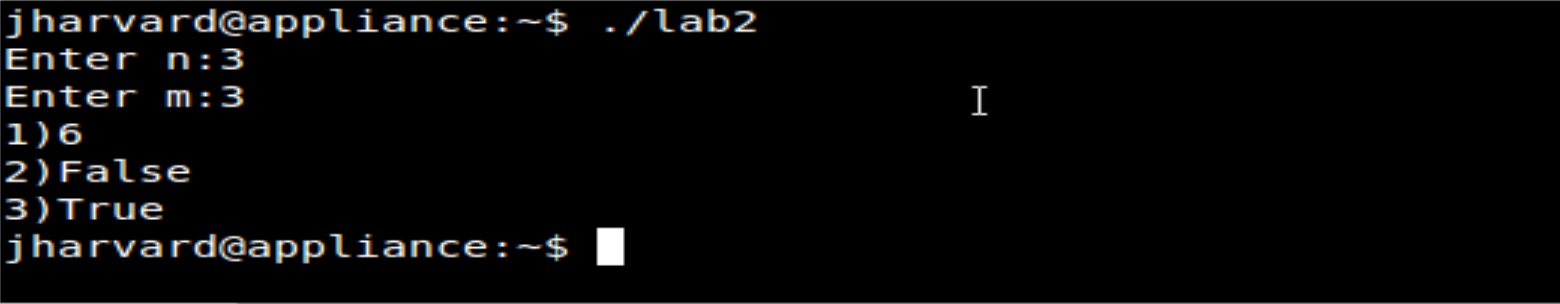
else

printf("3)False\n");

return 0;

}

**7. Результати роботи програми.**



1) n+++m= 6, при n=3, m=3

2) m-->n= false, при n=3, m=3

3) n-->m= true, при n=3, m=3

**8. Пояснення результатів.**

1. спочатку змінні додаються, а потім, оскільки ++ знаходиться після n, n збільшується на одиницю.

2. Оскільки тепер n=4, a m=3, то умова m>n є неправдою, а отже набуває значення

false.

3. Оскільки тепер n=4, a m=2, то умова n>m є правдою, а отже набуває значення

true, а оскільки -- знаходиться після m, m зменшується на одиницю.