

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() {

    float b=0.0001000, c, a=1000.0000000,k;

    c= pow(a-b, 3);
    printf("%f\n",c); //For checking definition of variables

    c-= pow(a, 3);
    printf("%f\n",c);

    k=pow(b, 3);
    printf("%f\n",k);

    k -= 3*a*pow(b, 2);
    printf("%f\n",k);

    k -= 3*b*pow(a, 2);
    printf("%f\n",k);

    c/=k;
    printf("%f", c);
    return 0;
}
```

3. Результати роботи програми для даних типу float.

1.066667

4. Результати роботи програми для даних типу double.
1.000000

5. Пояснення результатів.

При $a=1000$ і $b=0.0001$ рівняння
$$\frac{(a-b)^3 - (a^3)}{b^3 - 3ab^2 - 3a^2b} \approx 1$$

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 степеня, при якому вона ставала надто малою для збереження у **float**.

6. Програма розв'язку завдання2.

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

```
int main(){
```

```
    int a, m=3, n=2;
```

```
    //task 1
```

```
    a= n++*m;
```

```
    printf("n=%d\n",n);//checking n after the operation
```

```
    printf("%d\n", a);
```

```
    printf("Task 2\n");//task 2
```

```
    a= n++<m;
```

```
    printf("n=%d\n",n);//checking n after the operation
```

```
    printf("%d\n", a);
```

```
    printf("Task 2\n");//task 3
```

```
    a=m-- >m;
```

```
    printf("m=%d\n",m);//checking m after the operation
```

```
    printf("%d\n", a);
```

```
    return 0;
```

```
}
```

7. Результати роботи програми.

```
n=3
6
Task 2
n=4
0
Task 2
m=2
1

Process returned 0 (0x0)   execution time : 0.006 s
Press any key to continue.
```

- 1) $n++ * m = 6$, при
- 2) $n++ < m = 0$ (false), при $n=3$, $m=3$
- 3) $m-- > m = 1$ (true), при $m=3$

8. Пояснення результатів.

1. Першу формулу можна переписати наступним чином

$$a = n++ * m \mapsto \left(\begin{array}{l} a = n * m; \\ n = n + 1; \end{array} \right)$$

Тобто спочатку змінні перемножуються, а потім, оскільки ++ в постпозиції, n збільшується на одиницю.

2. Другу формулу можна переписати наступним чином

$$n++ < m \mapsto \left(\begin{array}{l} n < m; \\ n = n + 1; \end{array} \right)$$

Оскільки $n=m$, то умова $n < m$ є неправдою, а отже набуває значення 0(false).

3. Третю формулу можна переписати наступним чином

$$m-- > m \mapsto \left(\begin{array}{l} 3 > m; \\ m = m - 1; \end{array} \right)$$

Початкове значення $m=3$, тому після того, як значення змінної вставлено в формулу, оскільки -- в постпозиції, $m = m - 1$.

Прогрес в CS50

Лекція 0-1 (3/3) 100%

Бали за практику: 3/3

Лекція 0-2

Немає балів за виконані у цьому розділі завдання

Путівник по Scratch

Немає балів за виконані у цьому розділі завдання

Завдання 0

Немає балів за виконані у цьому розділі завдання

Тиждень 1

Лекція 1-1

Немає балів за виконані у цьому розділі завдання

Лекція 1-2

Немає балів за виконані у цьому розділі завдання

Короткі відео

Немає балів за виконані у цьому розділі завдання

Віртуальна лабораторія

Немає балів за виконані у цьому розділі завдання

Путівник по C

Немає балів за виконані у цьому розділі завдання

**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
Петров Дмитро
Викладач:
Гасько Р.Т.

Львів – 2018 р.