Звіт

з лабораторної роботи №1

дисципліни алгоритмізація і програмування ч.1

Виконав студент

Групи КН-112:

Тиський Святослав

Перевірила:

Старший викладач

кафедри СШІ

Мочурад Л.І.

2019р.

# Лабораторна робота №1.

**Варіант:28**

# Тема: "Знайомство з С. Виконання програми простої структури"

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

**Постанова 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.

при а=1000, b=0,001.

1. Обчислити значення виразів. Пояснити отримані результати.

n---m

m--<n

n++>m

**Програмна реалізація:**

* 1. **Тип даних float.**

#include <iostream>

using namespace std;

void main() {

int i;

float

a = 1000,

b = 0.001,

c = pow(a + b, 3),

d = (pow(a, 3) - (3 \* pow(a, 2)) \* b),

e = c + d,

f = 3 \* a \* pow(b, 2) + pow(b, 3);

printf("%.25f\n", f);

i = sizeof(f);

printf("%d\n", i);

};

**Результат:**

****

**Завдання №1.2 Типи даних double.**

#include <iostream>

using namespace std;

void main() {

int a = 1000, i;

double

b = 0.001,

c = pow(a + b, 3),

d = (pow(a, 3) - (3 \* pow(a, 2)) \* b),

e = c + d,

f = 3 \* a \* pow(b, 2) + pow(b, 3);

printf("%.25f\n", f);

i = sizeof(f);

printf("%d\n", i);

};

**Результат:**

****

**Пояснення:**

Порівнюючи два варіанти відповідей, можна помітити, що для однакової умови програма видала різні відповіді. Також за допомогою команди *sizeof* у кожному з варіантів я зміг визначити розмір кінцевої функції *f.* У варіанті 1, у якому я використовував тип даних *float,* функція *f* займає 4 байта, так як у варіанті 2 вона займає 8 байт. Це відбувається тому, що у типа даних *double* діапазон значень більший тому і результат точніший але й ,відповідно, більше затратний.

**Завдання №2**

#include<stdio.h>

#include<locale.h>

int main() {

int m, n, result1, result2, result3;

setlocale(LC\_ALL, "Ukrainian");

printf("Введiть m ");

scanf\_s("%d", &m);

printf("\nВведiть n ");

scanf\_s("%d", &n);

result1 = n---m;

printf("n---m = %d n=%d m=%d ", result1,n,m);

result2 = m--<n;

printf("\nm--<n = %s n=%d m=%d", result2 ? "true" : "false",n,m);

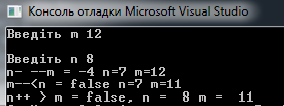
result3 = n++ > m;

printf("\nn++ > m = %s", n=%d m=%d", result3 ? "true" : "false",n,m);

return 0;

};

**Результат:**

****

**Пояснення:**

*Вираз 1:*

*--m* – інкремент стоїть перед числом, отже виконується до обчислення значення виразу.

*Вираз 2:*

*m--* -інкремент стоїть після числа, отже виконується після обчислення значення виразу і не впливає на нього.

*Вираз 3:*

n++ -інкремент стоїть після числа, отже виконується після обчислення значення виразу і не впливає на нього.

**Висновок:** на цій лабораторній роборі я ознайомився з середовищем програмування, створення, відлагодження й виконання простої програми, що містить ввід/вивід інформації й найпростіші обчислення.