Лукевич Володимир Андрійович Група РЕ-12 варіант 8

Створення програми з викорстанням циклів, перевірка введених значень на їхню правильність.

Програма складається із з циклів (while, do, for, if, else, else if)

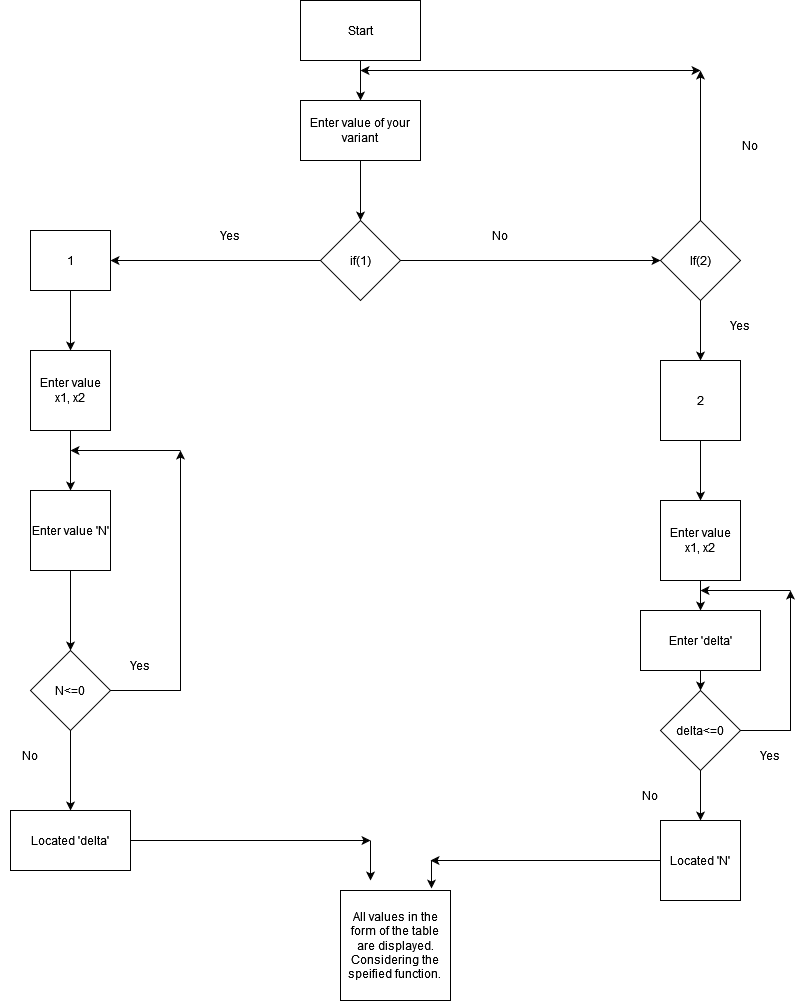
Використано бібліотеки:

**Stdio.h -**заголовний файл стандартної бібліотеки мови С, містить визначення макросів, контант, оголшених функцій і типів призначених дл виконання операцій веденя і виведення.

Stdlib.h – містить у собі функцій, що займаються виділенням памєяті конролем процесу виконання програм, перетворенням

Типів та інші.

conio.h – титульний файл використовується у компіляторах, що працюють на операційних системах МS-DOS( для створення тестового інтерфейсу користувача)



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

int main()

{

double x1, x2, delta, x, y, y2;

unsigned int variant, N, n;

do {

printf("Enter variant (1 or 2): ");

scanf("%u", &variant);

printf("var= %u", variant);

if (variant != 1 && variant != 2)

{

fprintf(stdout, "%s\n", "\t\n Wrong variant");

}

} while (variant != 1 && variant != 2);

if (variant == 1)

{

printf("type initial argument value x1 ");

scanf("%lf", &x1);

printf("type last argument value x2 ");

scanf("%lf", &x2);

printf("type the value of the number of points N- ");

scanf("%u", &N);

do

{

if (N != 0)

{

delta = (x2 - x1) / (N - 1);

}

else

{

printf("type the value of the number of points N-(can't be zero)");

scanf("%u", &N);

}

} while (N == 0);

system("cls");

y = 2 \* x1;

y2 = 2 \* x2;

//printf("You type: %s %lf %s %lf %s %u\n", "\ initial argument value x1 -", x1, "\n last argument value x2- ", "\ value of the number of points N -", N);

//scanf("%u", &N);

//printf("F(x)=2\* x1, F(x)=2 \* x2\n");

//printf("-------------------------");

//printf("N | X | f(X)");

printf("You type:: %s %lf %s %lf %s %u\n", "\n initial argument value x1 - ", x1, "\n last argument value x2-", x2, "\n value of the number of points N -", N);

printf("F(x) = 2 \* x1, F(x) = 2 \* x2\n");

printf("-----------------------------------------\n");

printf(" N | X | F(X) \n"

"-------------+--------------+-------------\n");

printf("%s %u %s %lf %s %lf %s ", " ", N, " |", x1, "|", y, "|", "\n"

"-------------+--------------+-------------\n");

printf(" | | \n"

"-------------+--------------+-------------\n");

printf("%s %u %s %lf %s %lf %s ", " ", N, " |", x2, "|", y2, "|", "\n"

"-------------+--------------+-------------\n");

}

else if (variant == 2)

{

printf(" type initial argument value x1- ");

scanf("%lf", &x1);

printf(" type last argument value x2- ");

scanf("%lf", &x2);

printf(" type the value of the number of points delta- ");

scanf("%lf", &delta);

do {

if (delta != 0)

{

N = 1 + (x2 - x1) / delta;

}

else

{

printf(" delta can't be zero ");

printf(" delta- ");

scanf("%lf", &delta);

}

} while (delta == 0);

system("cls");

printf(" You type: %s %lf %s %lf %s %lf\n", "\n initial argument value x1 - ", x1, "\n last argument value x2- ", x2, "\n step change argument delta- ", delta);

y = 2 \* x1;

y2 = 2 \* x2;

printf("-----------------------------------------\n");

printf(" N | X | F(X) \n"

"-------------+--------------+-------------\n");

printf("%s %u %s %lf %s %lf %s ", " ", N, " | ", x1, "| ", y, "|", "\n"

"-------------+--------------+-------------\n");

printf(" | | \n"

"-------------+--------------+-------------\n");

printf("%s %u %s %lf %s %lf %s ", " ", N, " | ", x2, "| ", y2, "|", "\n"

"-------------+--------------+-------------\n");

}while (variant != 1 && variant != 2);

//printf("x1=");

//scanf("%lf", &x1);

//printf("x2=");

//scanf("%lf", &x2);

///if (variant == 1) {

// printf("N=");

// scanf("%u", &N);

// }

//else {

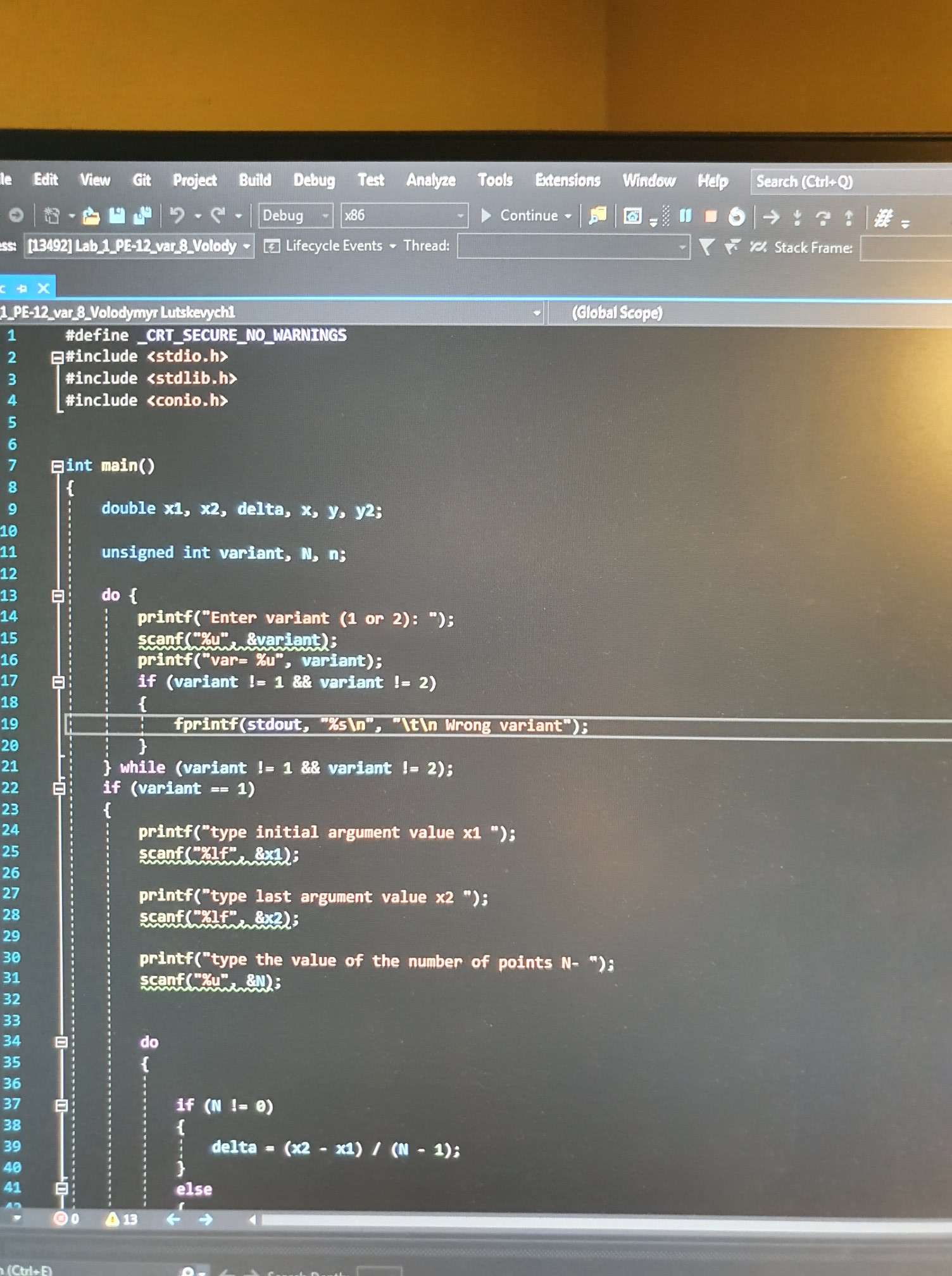
// printf("delta");

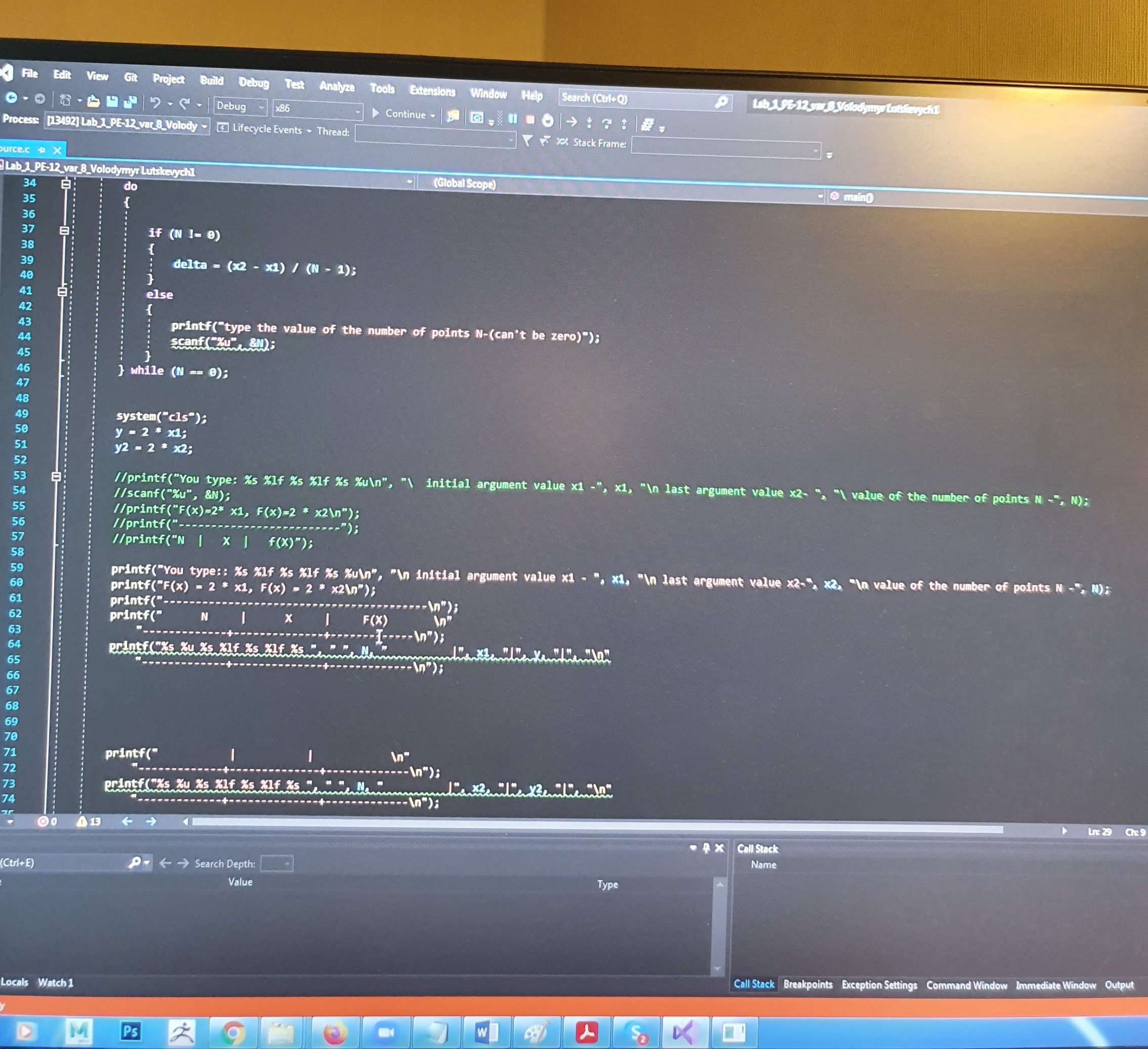
// scanf("%lf", &delta);

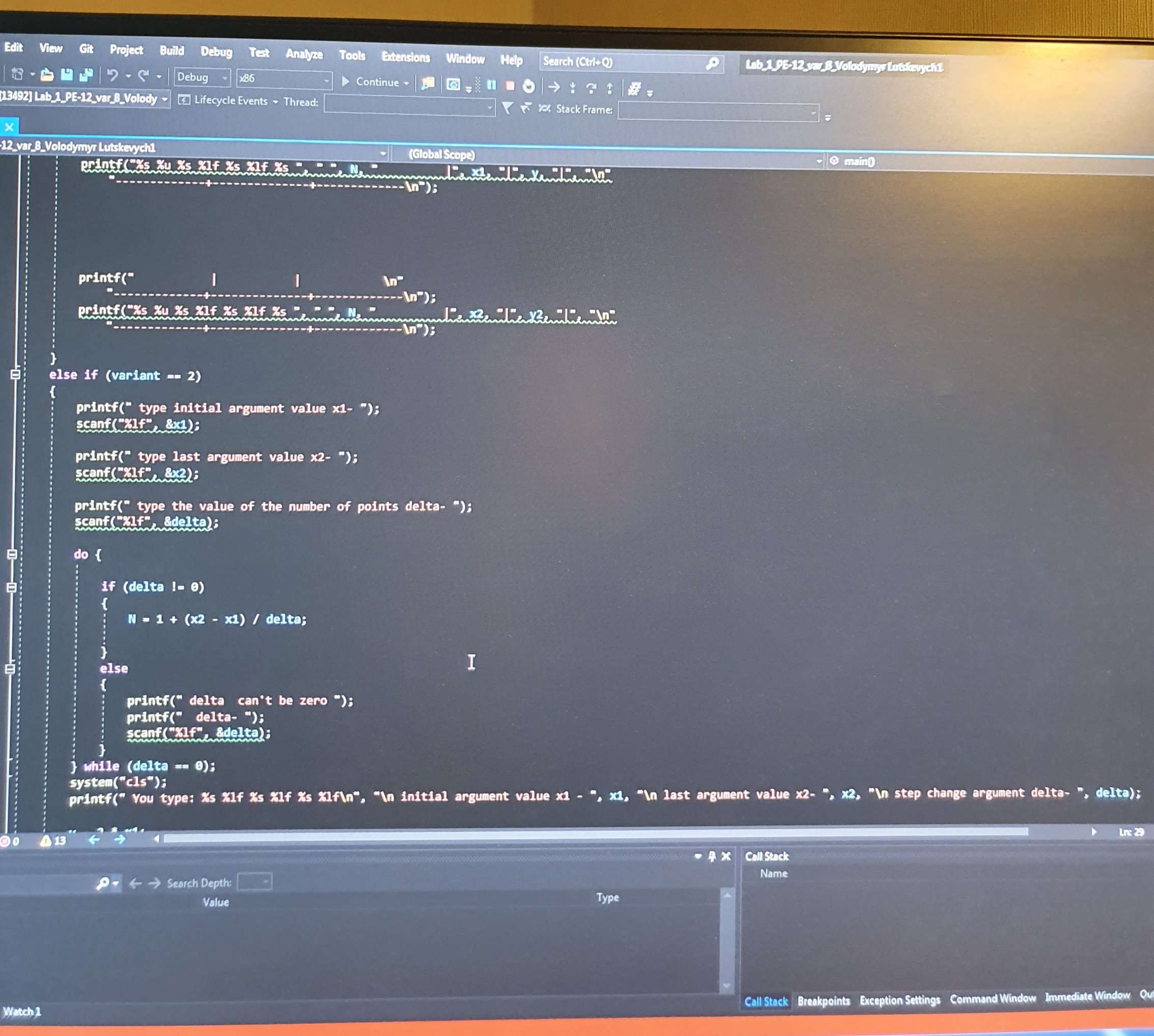
// }

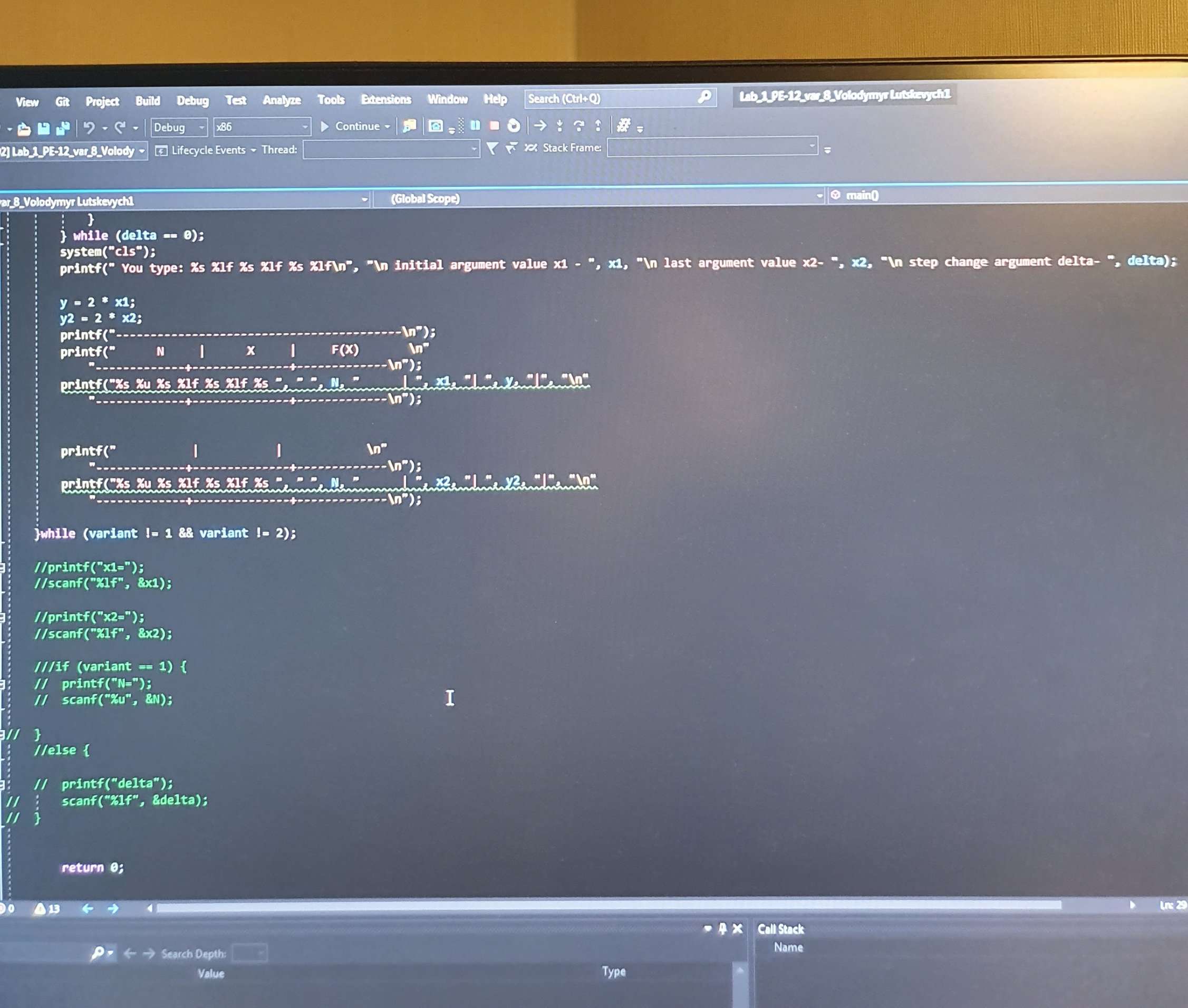
return 0;

}









Висновок: При використанні оператора ‘цикл’ оптимізується та вдосконавлюється програма.

Для розв’язання поставленої задачі необхідно створити алгоритм в якому покроково розписувати кожну дію і тільки потім писати програмний код. Весь час перевіряти почастинно програму для запобігання помилок та їхньої наявність.