Номер 1

1. Подсчитать число точек, находящихся внутри круга радиусом r с центром в начале координат; координаты заданы массивами X(10), Y(10).(В круге с радиусом 5)
2. float radius, promezutok=0 – потому, что эти переменные могут быть дробными

int proverka, colvo=0 – первая переменная нужна для проверки максимального целочисленного значения координаты, а вторая – просто количество точек, то есть обе не могут быть дробными.

I!=1

i=0(1)radius

promezutok=0 colvo=0

proverka

i=1

radius= случайное число

radius

Radius – chislo?

Да

Нет

Нет

Да

начало

radius

I=0

radius>=0 and radius<=5

J=0(1)radius

(i\*i+j\*j)<radius^2

Нет

Да

colvo=colvo+1

j

i

proverka=radius/1;

promezutok=(proverka\*2-1)\*2-1+4

colvo=(colvo\*4)+promezutok

Kolichestvo tochek v krugu takogo radiusa

конец

#include <iostream>

#include <stdio.h>

using namespace std;

void colvo(float radius) - //считает количество точек

{

float promezutok=0;

int proverka,colvo=0;

for(int i=1;i<radius;i++) {

for(int j=1;j<radius;j++) {

if((i\*i+j\*j)<(radius\*radius)) { //теорема пифагора

colvo=colvo+1; //количество точек, не лежащих на осях координат

} else {continue;}

}

}

proverka=radius/1; // последняя по индексу точка, лежащая на оси координат

//cout << proverka << endl;

promezutok=(proverka\*2-1)\*2-1+4; - //количество точек, лежащих на осях координат

colvo=(colvo\*4)+promezutok; //общее количество точек, находящихся в кругу

printf("Kolichestvo tochek v krugu takogo radiusa: %d",colvo);

}

void proverka\_vvoda(float radius) { //проверка введенных данных

for(int i=0;i!=1;) {

if (radius>=0 and radius<=5) { //проверка на ввод числа

i=1;

} else {

radius= 0 + rand() % 5; //задание случайного радиуса (работает только при вводе буквы)

cout << "Povtorite vvod!Pri vvode bukvi radius budet sgenerirovan sluchaino" << endl;

scanf("%f",&radius);

}

}

cout << "Radius:" << radius << endl;

colvo(radius); //переход на функцию поиска количества точек

}

int main()

{

float radius;

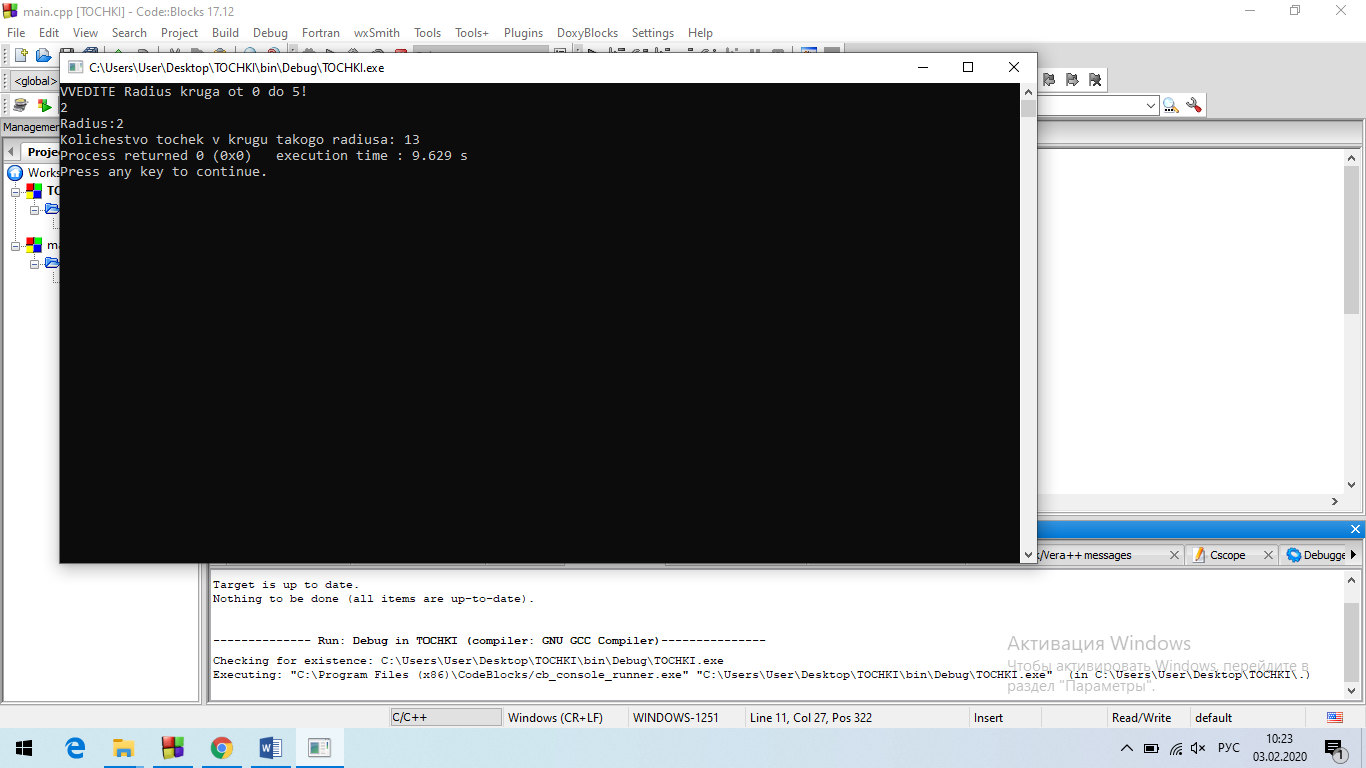
cout << "VVEDITE Radius kruga ot 0 do 5!" << endl;

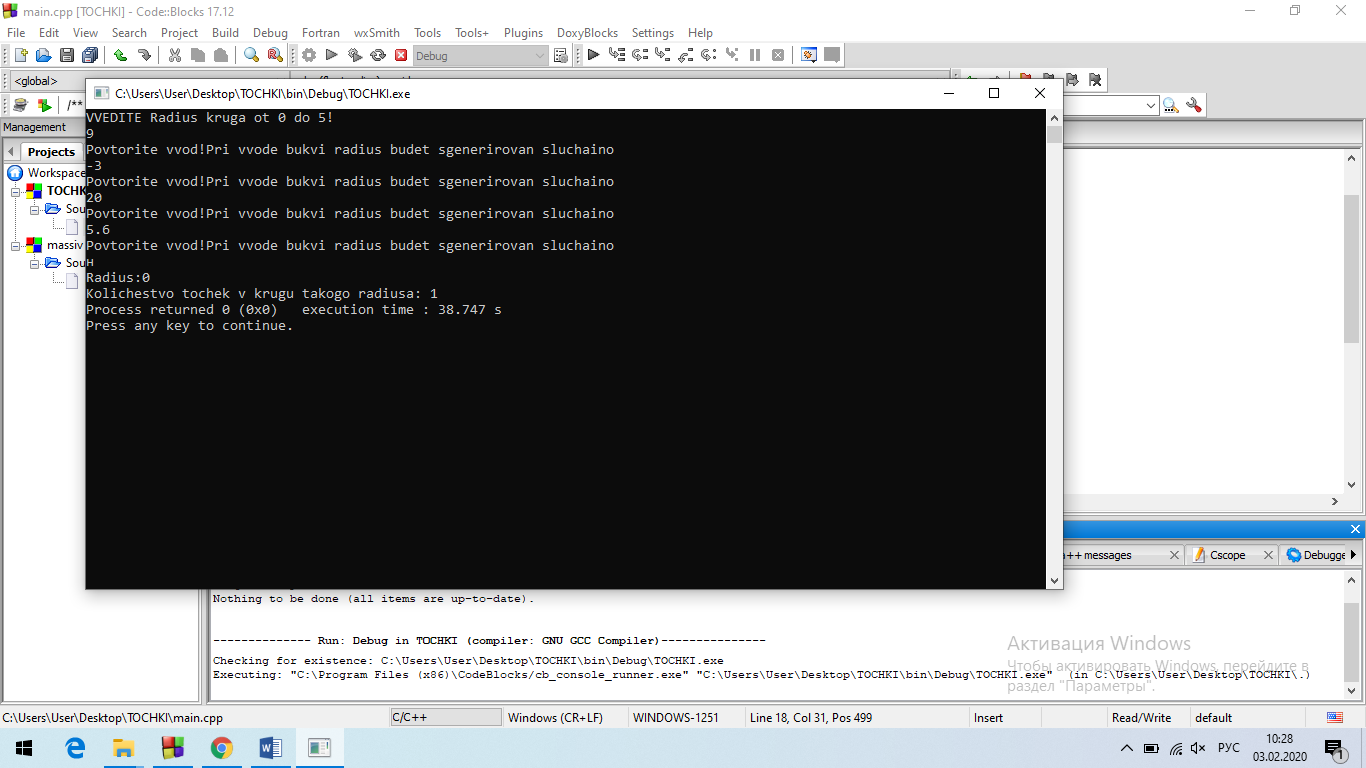
scanf("%f",&radius);

proverka\_vvoda(radius); //переход к функции проверки введенных данных

return 0;

}





1.Переписать положительные элементы массива x(10) и y(14) в массив z подряд.

2. int N=0,X[10],Y[14] – потому что в этой программе достаточно целочисленных данных и они мало весят.

Номер 2

N=0, Z[razmernost]

j

N++

Y[j]

j

Y[j] = -5 + rand() % (5 + 10)

J=0(1)14

N++

X[j]

J=0(1)10

X[j] = -5 + rand() % (5 + 10)

начало

N=0, X[10], Y[14]

J=0(1)10

X[j]>=0

Нет

Да

Z[N]=X[j]

N++

j

J=0(1)14

Y[j]>=0

Нет

Да

Z[N]=Y[j]

N++

j

Z

конец

#include <iostream>

#include <stdio.h>

using namespace std;

void zapis\_v\_massiv(int X[],int Y[],int razmernost)

{

int Z[razmernost],N=0;

for (int j = 0; j < 10; ++j){ //перенос положительных элементов из Х

if (X[j]>=0) {

Z[N]=X[j];

N++;

} else {continue;}

}

for (int j = 0; j < 14; ++j){ //перенос из У

if (Y[j]>=0) {

Z[N]=Y[j];

N++;

} else {continue;}

}

for (int j = 0; j < razmernost; ++j){

printf(" %d",Z[j]);

}

}

int main()

{

int N=0,X[10],Y[14];

for (int j = 0; j < 10; ++j){ //ввод массива рандомом

X[j] = -5 + rand() % (5 + 10);

printf(" %d",X[j]);

if (X[j]>=0) {

N++;

} else {continue;}

}

cout<<"\n"<<endl;

for (int j = 0; j < 14; ++j){

Y[j] = -7 + rand() % (5 + 10);

printf(" %d",Y[j]);

if (Y[j]>=0) {

N++;

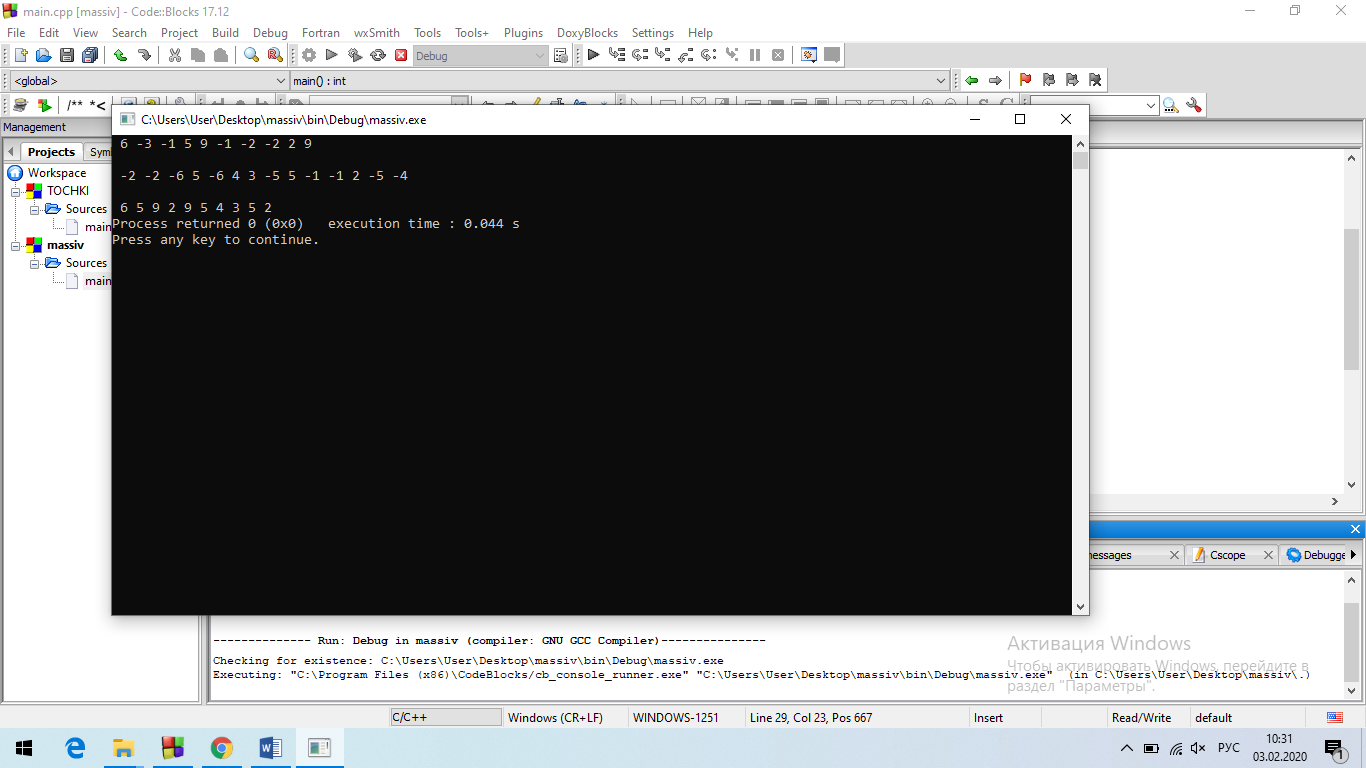
} else {continue;}

}

cout<<"\n"<<endl;

zapis\_v\_massiv(X,Y,N);

}



Этапа тестирование нет, потому что программ построена так, чтобы работать для любых массивов такой размерности. Массивы задаются рандомом, который при перезапуске программы не меняется, поэтому, если продолжать тестирование, то результат меняться не будет.

Ошибки в моем коде:

1 – неправильная разметка

2 – Неправильный тип данных у вывода

3 – Ориентация не на ту переменную в цикле.

4 – Вынос важного действия в комент.

5 – Неправильный внос рандома при введении символьных данных.

6 – Неправильное использование переменной в функции.

7 – Ошибка в цикле поиска ошибок ввода.

8 – Тип данных при вводе данных с клавиатуры.

9 – Неправильные синтаксис функции scanf.

10 – Ошибка в начальном значении переменной количества точек.

**Ошибки в кодах, предоставленных моим партнером:**

**Код номер 1:**

1)в строчке подсчета длины стороны треугольника:

return pow((Xtochki2-Xtochki1)+pow(Ytochki2-Ytochki1,2),0.5); - вот так она была записана в коде, который мне прислали

return pow(pow(Xtochki2-Xtochki1,2)+pow(Ytochki2-Ytochki1,2),0.5); - а вот так должно быть (возможно это 2 ошибки в одной строчке)

2)В строчке ввода массивов для координат (другое расширение):

char XA[5],XB[5],XC[5],YA[5],YB[5],YC[5],perimetr; - такой вид изначален

int XA[5],XB[5],XC[5],YA[5],YB[5],YC[5],perimetr; - а вот так должно быть (ещё можно использовать float)

3) Подсчет периметра:

perimetr=!Length(XB[n],YB[n],XA[n],YA[n])+Length(YC[n],YC[n],XA[n],YA[n])+Length(XB[n],YB[n],XC[n],YC[n]); - не нужен здесь этот восклицательный знак

4) Вывод периметра:

printf("\t\t\tPerimetr treugolnika #%c = %0.1f\n",n,perimetr); - не верно

printf("\t\t\tPerimetr treugolnika #%d = %0.1f\n",n,perimetr); - верно (но это повтор второго пункта, то есть возможно эти 2 ошибки на самом деле одна)

**Код номер 2:**

1)Начало цикла на заполнение матрицы:

for(int i=0;i<n;i--) - тут должно быть for(int i=0;i<n;i++)

2)Там же, в цикле матрицы (возможно это 1 ошибка, а не 2):

cout<<arr[i][-j]<<"\t"; - а должно быть cout<<arr[i][j]<<"\t";

3)Функция с поиском минимально элемента, внутренний цикл пробега:

for (int j = 1; j > m; j++), а должно быть for (int j = 1; j < m; j++)

4) if(minimum < arr[i][j]) – ошибка внутри внутреннего цикла из пункта 3, так должно быть if(minimum > arr[i][j])

5) a[i-2] = new int[4]; //Создание колонок - в этой строчке -2 не нужно.

Всего получилось найти 9 ошибок (в зависимости от трактовки от 7 до 10).