



// ConsoleApplication13.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

double Str(double a, double b, double c)

{

double p = (a + b + c) / 2;

return sqrt(p\*(p-a)\*(p-b)\*(p-c));

}

double Spr(double a, double b)

{

return a\*b;

}

int main()

{

double a, b, c, d, S;

cout << "a="; cin >> a;

cout << "b="; cin >> b;

cout << "c="; cin >> c;

cout << "d="; cin >> d;

S = Str(d, a, c) + Spr(a, b);

cout << "S=" << S << endl;

system("pause");

return 0;

}

// ConsoleApplication13.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//Дано 4 дійсних числа, з"ясувати скільки серед них дорівнюють нулю

int getZeroCount(double a, double b, double c, double d)

{

int count = 0;

if (a == 0) count++;

if (b == 0) count++;

if (c == 0) count++;

if (d == 0) count++;

return count;

}

int main()

{

double a, b, c, d;

cout << "a="; cin >> a;

cout << "b="; cin >> b;

cout << "c="; cin >> c;

cout << "d="; cin >> d;

cout << " Count =" << getZeroCount(a,b,c,d) << endl;

system("pause");

return 0;

}

// ConsoleApplication13.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//Дано масив цілих чисел. З"ясувати що більше по модулю сума додатніх чи сума від"ємних

int getPositiveSum(int \*a, int n) //Знаходить суму додатніх

{

int s = 0;

for (int i = 0; i < n; i++)

{

if (a[i] > 0) s += a[i];

}

return s;

}

int getNegativeSum(int \*a, int n) //Знаходить суму від"ємних

{

int s = 0;

for (int i = 0; i < n; i++)

{

if (a[i] < 0) s += a[i];

}

return s;

}

int main()

{

int n;

cout << "n="; cin >> n;

int \*a = new int[n];

for (int i = 0; i < n; i++)

{

a[i] = rand() % 50 - 25;

}

int pS = getPositiveSum(a, n);

int nS = getNegativeSum(a, n);

if (pS>-nS) //if(getPositiveSum(a, n) > -getNegativeSum(a, n))

{

cout << "pS> | nS |" << endl;

}

else

{

cout << "| nS |> pS " << endl;

}

system("pause");

return 0;

}



