**Question 1:**

Cho một cấu trúc sau:

typedef struct

{

int m\_integer1;

float m\_float;

bool m\_bool;

int m\_integer2;

char m\_char;

char\* m\_string;

} Packing;

Hãy cho biết size của Packing có thể optimize xuống nhỏ nhất là bao nhiêu (sử dụng default packing)?

(Solution: 20 bytes)

**Question 2:**

Trong hệ tọa độ (x,y) chúng ta có :

* Một đường tròn với tâm là (X0, Y0) và bán kính R
* Một đường thẳng với phương trình y = A\*x + B

Hãy lập trình để tìm điểm giao nhau giữa đường tròn và đường thẳng

**Input**: X0, Y0, R, A, B (là các số nguyên và R > 0)

**Output**:

*Dòng 1* : In ra YES/NO

(YES nếu có giao nhau và NO nếu không)

*Dòng 2* : (X1,Y1) (X2,Y2)

(In ra tọa độ của giao điểm và sắp theo thứ tứ tăng dần của x và làm tròn 3 chữ số sau dấu phẩy. Nếu không giao nhau thì không in dòng 2, nếu chỉ cắt tại 1 điểm thì chỉ in ra (X1,Y1))

**Ví dụ 1:**

*Input: 1 3 4 -2 5*

*Output:*

*YES*

*(-2.2,9.399)(1,3)*

**Solution:**

Solution :

#include "stdafx.h"

#include <iostream>

using namespace std;

float doRoundFoatAfterComma(float \_number, int \_n\_AfterComma)

{

long \_number\_adjust\_rounded = long(\_number \* pow(10, \_n\_AfterComma));

return \_number\_adjust\_rounded / (pow(10, \_n\_AfterComma));

}

int main()

{

int X0, Y0, A, B, R;

cout << "Input the coordinate of center of circle : (X0, Y0), radius of circle (R), and paramater (A, B) of equation of straight-line";

cout << "\nInput following the order : X0 Y0 R A B : ";

cin >> X0 >> Y0 >> R >> A >> B;

if (R <= 0)

{

//Wrong input for R since R has to > 0

return -1;

}

// The equation of circle : (x - X0)^2 + (y - Y0)^2 = R^2 (1)

// Replace y = A\*x + B to (1) to have quadratic equation

// (x - X0)^2 + (A\*x + B - Y0)^2 = R^2 ---> is same to

// x^2\*(1 + A^2) + x\*(2\*A\*Y0 - 2\*X0 - 2\*A\*B) + (X0^2 + B^2 + Y0^2 - 2\*B\*Y0 - R^2) = 0

// Let's replace m\_coefficient\_x\_2 = 1 + A^2, m\_coefficient\_x\_1 = 2\*A\*Y0 - 2\*X0 - 2\*A\*B, m\_coefficient = X0^2 + B^2 + Y0^2 - 2\*B\*Y0 - R^2

long m\_coefficient\_x\_2 = 1 + A\*A;

int m\_coefficient\_x\_1 = 2\*A\*Y0 - 2\*X0 - 2\*A\*B;

long m\_coefficient = X0\*X0 + B\*B + Y0\*Y0 - 2\*B\*Y0 - R\*R;

//Check how to solve quadratic equation from: https://en.wikipedia.org/wiki/Quadratic\_equation

//Check m\_coefficient\_x\_1\*m\_coefficient\_x\_1 - 4\*m\_coefficient\_x\_2\*m\_coefficient

long m\_delta\_coefficient = m\_coefficient\_x\_1\*m\_coefficient\_x\_1 - 4 \* m\_coefficient\_x\_2\*m\_coefficient;

if (m\_delta\_coefficient < 0)

{

cout << "\nNO";

}

else if (m\_delta\_coefficient == 0)

{

float m\_x\_intersect = doRoundFoatAfterComma((-m\_coefficient\_x\_1) / (2 \* m\_coefficient\_x\_2),3);

float m\_y\_intersect = doRoundFoatAfterComma(A\*m\_x\_intersect + B,3);

cout << "\nYES";

cout << "\n(" << m\_x\_intersect << "," << m\_y\_intersect << ")";

}

else

{

float m\_delta\_coefficient\_SQRT = sqrt(m\_delta\_coefficient);

float m\_x1\_intersect = doRoundFoatAfterComma((-m\_coefficient\_x\_1 + m\_delta\_coefficient\_SQRT) / (2 \* m\_coefficient\_x\_2),3);

float m\_y1\_intersect = doRoundFoatAfterComma(A\*m\_x1\_intersect + B,3);

float m\_x2\_intersect = doRoundFoatAfterComma((-m\_coefficient\_x\_1 - m\_delta\_coefficient\_SQRT) / (2 \* m\_coefficient\_x\_2),3);

float m\_y2\_intersect = doRoundFoatAfterComma(A\*m\_x2\_intersect + B,3);

cout << "\nYES";

if (m\_x1\_intersect <= m\_x2\_intersect)

{

cout << "\n(" << m\_x1\_intersect << "," << m\_y1\_intersect << ")";

cout << "(" << m\_x2\_intersect << "," << m\_y2\_intersect << ")";

}

else

{

cout << "\n(" << m\_x2\_intersect << "," << m\_y2\_intersect << ")";

cout << "(" << m\_x1\_intersect << "," << m\_y1\_intersect << ")";

}

}

cout << "\n";

system("pause");

return 0;

}

**Test case 1:**

Input: 2 0 1 0 1

Output:

YES

(2,1)

**Test case 2:**

Input: **2 3 2 1 3**

YES

(0,3)(2,5)

**Test case 3:**

Input: -3 -4 1 2 7

Output:

NO

**Test case 4:**

Input: **1 3 5 3 7**

Output:

YES

(-0.277,6.169)(2.877,15.631)

**Test case 5:**

Input: 7 -15 23 15 -30

Output:

YES

(-2.399,-65.985)(0.47,-22.95)

**Question 3:**

Trên màn hình hiển thị của một game có 2 điểm màu A và B với tọa độ lần lượt là A(x1, y1) và B(x2, y2), nối các điểm màu từ A tới B để tạo thành đoạn thẳng chúng ta thường thấy xuất hiện vấn đề Răng Cưa.

Input : nhập vào tọa độ của X

Output : xuất ra giá trị của Y mà khoảng cách từ điểm có tọa độ (X,Y) đến đường thẳng là ngắn nhất. Lưu ý khoảng cách là kiểu float và làm tròn 3 số sau dấu phẩy. Khoảng cách này có thể là 0 trong trường hợp (X,Y) nằm trên đường thẳng

**Question 4:**

Input : nhập vào 1 số nguyên dương N

Output : cho biết số nguyên tố lớn nhất là ước số của N

**Question 5:**

Graphics API nào dưới đây được sử dụng đầu tiên?

1. OpenGL
2. DirectX
3. Metal
4. Vukan

<https://en.wikipedia.org/wiki/DirectX>

<https://www.khronos.org/opengl/wiki/History_of_OpenGL>