

1. Код

```
/*
Координаты и радиус предполагаются целыми. A - центр одного из оснований конуса, B -
другого, R - радиуса основания.
Точка X лежит внутри цилиндра, если выполняются два условия:
1. точка X лежит между плоскостями оснований
2. расстояние от X до АВ меньше R

Первое равносильно тому, что два угла ХАВ и ХВА не меньше 0 и не больше  $\pi/2$ , или, что
тоже самое, скалярное произведение сторон угла (как векторов) больше 0.
Второе - рассматриваем треугольник АХВ, его площадь - это полсумма основания на высоту,
либо полсумма сторон на синус угла. Отсюда можно выразить высоту, которая и есть
расстояние от X до АВ.
*/

#include <iostream>

using namespace std;

struct point {
    int x;
    int y;
    int z;
};

struct cylinder {
    point a;
    point b;
    int radius;
};

void inputPoint(point &A) {
    cout << "Enter point coordinates. x, y, z:\n";
    cin >> A.x >> A.y >> A.z;
}

void printPoint(point A) {
    cout << "Point coordinates: " << A.x << " " << A.y << " " << A.z << endl;
}

void inputCylinder(cylinder& c) {
    cout << "Enter coordinates of A and B:\n";
    inputPoint(c.a);
    inputPoint(c.b);
    cout << "Enter radius of the base\n";
    cin >> c.radius;
}

void printCylinder(cylinder C) {
    cout << "";
}

int dotProduct(point a, point b, point c, point d) {
    //calculates dot product of vectors ab and cd
    return (b.x - a.x) * (d.x - c.x) + (b.y - a.y) * (d.y - c.y) + (b.z - a.z) * (d.z - c.z);
}
```

```

}

bool isInCylinder(point x, cylinder c) {
    if ((dotProduct(c.a, x, c.a, c.b) < 0) || (dotProduct(c.b, x, c.b, c.a) < 0)) {
        return false;
    }

    if ((dotProduct(c.a, x, c.a, x) * dotProduct(c.a, c.b, c.a, c.b) - dotProduct(c.a, x,
c.a, c.b) * dotProduct(c.a, x, c.a, c.b)) > c.radius * c.radius * dotProduct(c.a, c.b,
c.a, c.b)) {
        return false;
    }
    return true;
}

int main()
{
    point x = { 0,0,0 };
    cylinder c;
    cout << "Enter point and cylinder\n\n";
    //inputPoint(x);
    inputCylinder(c);
    string result = (isInCylinder(x, c)) ? "Yes" : "No";
    cout << "Q: Is the point inside of the cylinder?\nA: " << result;
    return 0;
}

```

2.

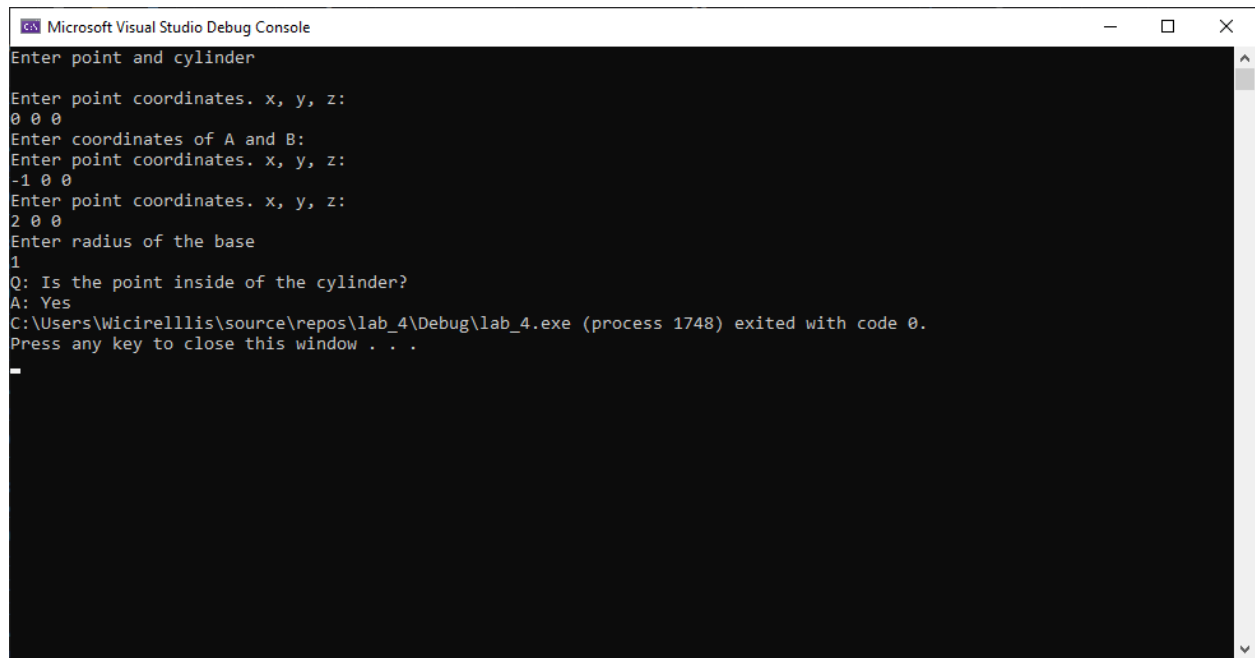
Проверим работу программы на каком-нибудь входе. Пусть ввод слеующий:

0 0 0

-1 0 0

2 0 0

1



```
Microsoft Visual Studio Debug Console
Enter point and cylinder
Enter point coordinates. x, y, z:
0 0 0
Enter coordinates of A and B:
Enter point coordinates. x, y, z:
-1 0 0
Enter point coordinates. x, y, z:
2 0 0
Enter radius of the base
1
Q: Is the point inside of the cylinder?
A: Yes
C:\Users\Wicirelllis\source\repos\lab_4\Debug\lab_4.exe (process 1748) exited with code 0.
Press any key to close this window . . .
```

Тогда результат на скрине. Можно понять, что это правда.

3.

Несколько скринов вывода

```
Microsoft Visual Studio Debug Console
Enter point and cylinder
Enter point coordinates. x, y, z:
10 10 1
Enter coordinates of A and B:
Enter point coordinates. x, y, z:
0 0 0
Enter point coordinates. x, y, z:
1 1 10
Enter radius of the base
2
Q: Is the point inside of the cylinder?
A: No
C:\Users\wicirelllis\source\repos\lab_4\Debug\lab_4.exe (process 1956) exited with code 0.
Press any key to close this window . . .
```

```
Microsoft Visual Studio Debug Console
Enter point and cylinder
Enter point coordinates. x, y, z:
7 7 -7
Enter coordinates of A and B:
Enter point coordinates. x, y, z:
1 1 -1
Enter point coordinates. x, y, z:
10 10 -10
Enter radius of the base
1
Q: Is the point inside of the cylinder?
A: Yes
C:\Users\wicirelllis\source\repos\lab_4\Debug\lab_4.exe (process 15316) exited with code 0.
Press any key to close this window . . .
```

4. Вопросы

1) Что такое структура? Какие объекты может содержать структура? Как происходит обращение к элементам структуры?

Структура – это тип пользовательский тип данных, несколько переменных собранных в один объект.

Элементы структуры – это базовые типы (int, double, str итд), другие структуры.

К элементам можно обращаться через точку и стрелочку. Пусть например есть структура, представляющая точку в R3

```
struct point {  
    int x;  
    int y;  
    int z;  
};
```

```
point myPoint;  
myPoint.x;
```

```
point* ptr_myPoint;  
ptr_myPoint->x;
```

myPoint – это инстанс структуры, и через точку можно получить первую координату. Разница между . и -> в том, что точка применяется к структуре, а стрелочка к указателю на структуру.

2) Возможна ли при инициализации структуры сразу присвоить ей значение? Приведите пример как, если возможно.

Да, возможно. Через равно и фигурные скобки. Например

```
point myPoint;  
myPoint.x = { 0,0,0 };
```