헤더파일

```
#ifndef POINT_H
#define POINT_H

struct Point
{
	double x;
	double y;
};

class PointSet
{
private:
	Point* points;
	int n;

public:
	PointSet(int n);
	~PointSet();


	void genPoints(double minX, double maxX, double minY, double maxY);
	double distance(const Point& p1, const Point& p2);
	void MinMaxDistance();
};
```

```
#endif // POINT_H
```

소스파일

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
#include "point.h"

using namespace std;

PointSet::PointSet(int n)
{
    this->n = n;
    points = new Point[n];
}

PointSet::~PointSet()
{
    delete[] points;
}

void PointSet::genPoints(double minX, double maxX, double minY, double maxY)
{
    srand(time(0)); // 랜덤 시드 초기화
    for (int i = 0; i < n; i++) {
            points[i].x = minX + (rand() / (RAND_MAX / (maxX - minX)));
            points[i].y = minY + (rand() / (RAND_MAX / (maxY - minY)));
        }
}

double PointSet::distance(const Point& p1, const Point& p2)
{
        return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y, 2));
}

void PointSet::MinMaxDistance()
{
        double minDist = distance(points[0], points[1]);
        double maxDist = minDist;
        Point minP1 = points[0], minP2 = points[1];
        Point maxP1 = points[0], maxP2 = points[1];

        for (int i = 0; i < n; i++)
{
            for (int j = i + 1; j < n; j++)
{
                    double dist = distance(points[i], points[j]);
                    if (dist < minDist)
{
                            minDist = dist;
```

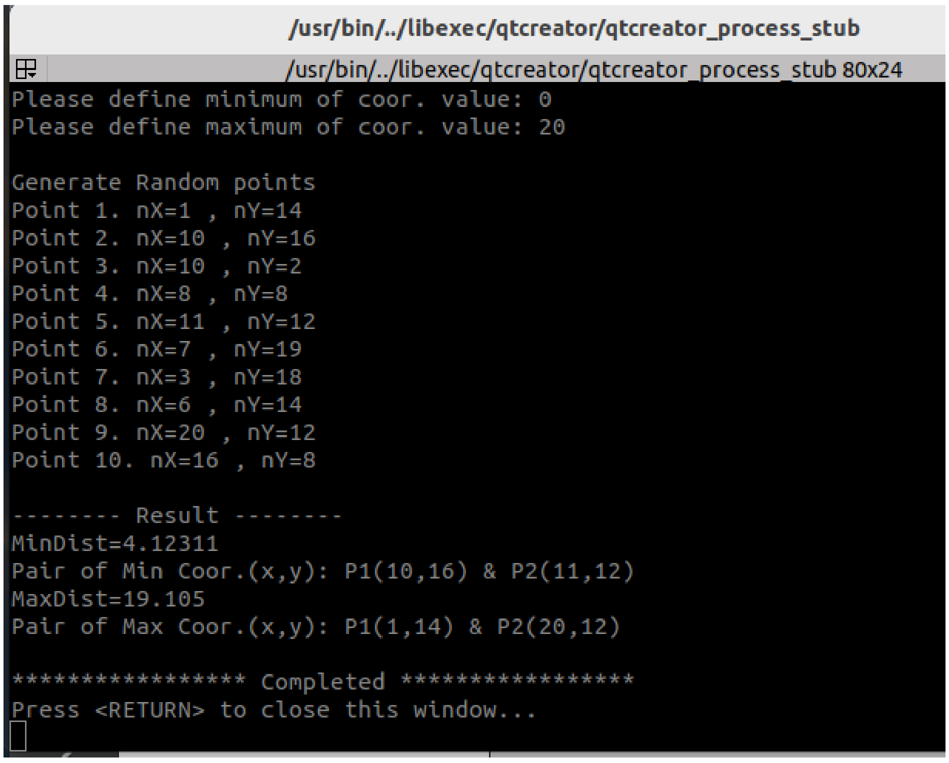
```
                minP1 = points[i];

                minP2 = points[j];

            }

            if (dist > maxDist)

{

                maxDist = dist;

                maxP1 = points[i];

                maxP2 = points[j];

            }

        }

    }


    cout << "최소 거리: " << minDist << " (" << minP1.x << ", " << minP1.y << ") 와 
(" << minP2.x << ", " << minP2.y << ")" << endl;

    cout << "최대 거리: " << maxDist << " (" << maxP1.x << ", " << maxP1.y << ") 와 
(" << maxP2.x << ", " << maxP2.y << ")" << endl;

}
```

메인파일

```
#include <iostream>

#include <vector>

#include <cmath>

#include <cstdlib>

#include <ctime>


using namespace std;


struct Point

{

	int x, y;

};


double calcDistance(const Point& p1, const Point& p2)

{

	return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y, 2));

}


int main()

{

	int numPoints, minCoord, maxCoord;


	cout << "********** HW 3 Point Distance Computation **********₩n₩n";
	cout << "Please define the number of points: ";
	cin >> numPoints;
	cout << "Please define minimum of coor. value: ";
	cin >> minCoord;
	cout << "Please define maximum of coor. value: ";
	cin >> maxCoord;

	srand(time(0));
	vector<Point> points(numPoints);
	for (int i = 0; i < numPoints; ++i)
	{
		points[i].x = rand() % (maxCoord - minCoord + 1) + minCoord;
		points[i].y = rand() % (maxCoord - minCoord + 1) + minCoord;
	}

	cout << "₩nGenerate Random points₩n";
	for (int i = 0; i < numPoints; ++i)
	{
		cout << "Point " << i + 1 << ". nX=" << points[i].x << " , nY=" << 
points[i].y << "₩n";
	}

	double minDist = calcDistance(points[0], points[1]);
	double maxDist = minDist;
```

```
Point minP1 = points[0], minP2 = points[1];
Point maxP1 = points[0], maxP2 = points[1];

for (int i = 0; i < numPoints; ++i)
{
      for (int j = i + 1; j < numPoints; ++j)
      {
            double dist = calcDistance(points[i], points[j]);
            if (dist < minDist)
            {
                  minDist = dist;
                  minP1 = points[i];
                  minP2 = points[j];
            }
            if (dist > maxDist)
            {
                  maxDist = dist;
                  maxP1 = points[i];
                  maxP2 = points[j];
            }
      }
}

cout << "₩n-------- Result --------₩n";
```

```
    cout << "MinDist=" << minDist << "₩n";
    cout << "Pair of Min Coor.(x,y): P1(" << minP1.x << "," << minP1.y << ") & P2(" << minP2.x << "," << minP2.y << ")₩n";
    cout << "MaxDist=" << maxDist << "₩n";
    cout << "Pair of Max Coor.(x,y): P1(" << maxP1.x << "," << maxP1.y << ") & P2(" << maxP2.x << "," << maxP2.y << ")₩n";

    cout << "₩n***************** Completed *****************₩n";
    cout << "Press <RETURN> to close this window...₩n";
    cin.get();
    cin.get();

    return 0;
}
```



/usr/bin/../libexec/qtcreator/qtcreator_process_stub

/usr/bin/../libexec/qtcreator/qtcreator_process_stub 80x24

```
Please define minimum of coor. value: 0
Please define maximum of coor. value: 20

Generate Random points
Point 1. nX=1 , nY=14
Point 2. nX=10 , nY=16
Point 3. nX=10 , nY=2
Point 4. nX=8 , nY=8
Point 5. nX=11 , nY=12
Point 6. nX=7 , nY=19
Point 7. nX=3 , nY=18
Point 8. nX=6 , nY=14
Point 9. nX=20 , nY=12
Point 10. nX=16 , nY=8

-------- Result --------
MinDist=4.12311
Pair of Min Coor.(x,y): P1(10,16) & P2(11,12)
MaxDist=19.105
Pair of Max Coor.(x,y): P1(1,14) & P2(20,12)

***************** Completed *****************
Press <RETURN> to close this window...
```