**Imperial College
London**

EE2-12 Software Engineering 2: Object-Oriented Programming

Week 3 – Lab*

Sahbi Ben Ismail (`s.ben-ismail@imperial.ac.uk`)

Duplicate points

Consider class `point` with the `<` and `==` operators:

(Note that that following functions should not be member functions of the `point` class).

Write a function `member_point_unordsearch` which takes in input a point object and a vector of points and returns the index of the point in the vector if it is found and the size of the vector otherwise.

Write a function `member_point_binsearch` analogous to the previous one but meant to be called when the vector is ordered and performing binary search.

(Optional) Write a function `member_point_sort` that sorts a vector of points (using bubble sort or otherwise)

Write a function `member_point` which takes in input a point object and a vector of points and one further parameter which states whether the vector is ordered or not. This parameter is set by default to `false`. The function returns the index of the point in the vector if it is found and the size of the vector otherwise.

If the parameter is set to `false` (or it is not passed), then `member_point` calls `member_point_unordsearch`.

---

*Lab content originally written by Max Cattafi.

If the parameter is set to `true`, then `member_point` calls `member_point_binsearch`.

Write a function `pointdup` which uses function `member_point`, takes in input two vectors and has a third vector as output parameter. This vector will contain all the elements of the first vector which are also in the second vector. Function `pointdup` takes one further parameter set by default to `false` expressing whether the second input vector is ordered or not, which should be in turn passed to function `member_point`.

Write a program to test the function which reads from a file a list of points and stores it into a vector of objects of type `point`, reads from another file another list of points and stores it into a second vector of objects of type `point`, calls `pointdup` on these two vectors and prints on a third file the result. The names of the files should be passed using the command line arguments (`argv`), as well as a parameter which states whether the second file is ordered or not.