ECS 40 **Final Handout B** **Spring 2015**

For this question, you will write a program that will read and display the results from the ECS 60 P5 testing script program stored in results.csv. (Please note that I have randomly changed the scores and names to yours to protect the innocent.) The students in 60 had to write programs that found paths across five city maps. The script would check the proposed path(s), and enter “Yes” or “No” depending on the validity of the paths proposed by the student programs. The time taken to run the program on each map followed the validity of the path. If the check was “No”, then the time was set to zero. The first line of results.csv contains the headings for the columns. Each succeeding lines contain the information for a student.

A student’s score was based on one of the times from five runs of their program. The student’s were assigned to either the Normal class or the Multiple class based on how their program performed. For programs that worked with all five maps, Students were in the Normal class, and the fastest time was used. However, the first map tested had only one path across the city. Many of their programs could only handle maps that had multiple paths across the city, and thus had “No” for the first map, and “Yes” for the other four maps. These students had to be graded based on the slowest time from the other four maps, and were placed in the Multiple class. If a student had “Nos” for any map other than the first, then their time was assigned 10000, and they were placed in the Normal class.

```
class Trial{
  bool valid;
  float time;
public:
  Trial(bool v, float t) : valid(v), time(t){}
  void print()const {if(valid) cout << "Y "; else cout << "N "; cout << time << "
";}
  bool operator<(const Trial &t2)const {return time < t2.time;}
  bool getValid()const {return valid;}
  float getTime() const {return time;}
}; // class Trial

class Normal{
protected:
  string name;
  multiset <Trial> trials;
public:
  Normal(const string &n, const vector <bool> &v, const vector <float> &t);
  void print() const;
  virtual float getTime() const;  // returns time selected
  const string& getName() const {return name;}
}; // class Normal

class Multiple : public Normal {
public:
  Multiple(const string &n, const vector <bool> &v, const vector <float> &t);
  float getTime() const;  // returns time selected
}; // class Multiple

typedef map <string, Normal*> ResultsMap;

class Results {
  ListNode <Normal> *head;    ResultsMap map;
public:
  Results() : head(NULL);
  ~Results();
  void print(const string &directory) const;
  Results& operator+= (Normal *n);
  friend istream& operator>> (istream &is, Results &result);
};
```

```
int main()
{
  Results results;
  string directory;
  ifstream inf("results.csv");
  inf >> results;

  do{
    cout << "Please enter a directory: ";
    cin >> directory;
    results.print(directory);
  } while (directory != "Done");
  return 0;
}
```

```
[ssdavis@lect1 final]$ cat results.csv
Directory,Name,Valid,Time,Valid,Time,Valid,Time,Valid,Time,Valid,Time
./linjo,Joseph Lin,Yes,4.05,Yes,4.45,Yes,4.44,Yes,4.39,Yes,4.44
./swensonb,Brian Swenson,No,0,No,0,No,0,No,0,No,0
./dubs,Andrey Dub,Yes,0.1,Yes,0.08,Yes,0.11,Yes,0.11,Yes,0.08
./shelby,John Shelby,Yes,0.49,Yes,1.38,Yes,1.58,Yes,1.54,Yes,1.27
./mccormir,Ryan McCormick,Yes,0.29,Yes,0.29,Yes,0.3,No,0,Yes,0.33
./tsaij,Jennifer Tsai,No,0,Yes,0.01,Yes,0,Yes,0,Yes,0.01
./changay,Andrew Chang,Yes,1.83,Yes,0.99,Yes,0.76,Yes,1.19,Yes,1.06
./chatterj,Surmi Chatterjee,No,0,Yes,0.04,Yes,0.04,No,0,Yes,0.04
./wongsih,Simon Wong,Yes,0.17,Yes,0.16,Yes,0.12,Yes,0.12,Yes,0.14
./chenwe,Wei-Chi Chen,No,0,Yes,42.22,Yes,43.34,Yes,45.69,Yes,45.34
./santosl,Liliana Santos,Yes,1.11,Yes,1.16,Yes,0.44,Yes,0.5,Yes,0.59
./puim,Dustin Puim,No,0,No,0,No,0,No,0,No,0
[ssdavis@lect1 final]$ a.out
Please enter a directory: ./linjo
Joseph Lin Time: 4.05  Runs: Y 4.05 Y 4.39 Y 4.44 Y 4.44 Y 4.45

Please enter a directory: ./swensonb
Brian Swenson Time: 10000  Runs: N 0 N 0 N 0 N 0 N 0

Please enter a directory: ./dubs
Andrey Dub Time: 0.08  Runs: Y 0.08 Y 0.08 Y 0.1 Y 0.11 Y 0.11

Please enter a directory: ./shelby
John Shelby Time: 0.49  Runs: Y 0.49 Y 1.27 Y 1.38 Y 1.54 Y 1.58

Please enter a directory: ./mccormir
Ryan McCormick Time: 10000  Runs: N 0 Y 0.29 Y 0.29 Y 0.3 Y 0.33

Please enter a directory: ./tsaij
Jennifer Tsai Time: 0.01  Runs: N 0 Y 0 Y 0 Y 0.01 Y 0.01

Please enter a directory: ./chenwe
Wei-Chi Chen Time: 45.69  Runs: N 0 Y 42.22 Y 43.34 Y 45.34 Y 45.69

Please enter a directory: Done
No such directory

[ssdavis@lect1 final]$
```