

Question 5

The following program is in [program.cpp](#):

```
#include <vector>
#include <cstdlib>
#include <iostream>
using namespace std;

main(int argc, char **argv)
{
    string m;
    string s;
    vector <string> v;
    int i;

    if (argc != 2) { cerr << "usage: p string\n"; exit(1); }
    m = argv[1];

    while (cin >> s) {
        if (s >= m) {
            v.push_back(s);
        } else {
            for (i = 0; i < v.size() && v[i] < m; i++) ;
            if (i == v.size()) {
                v.push_back(s);
            } else {
                v.push_back(v[i]);
                v[i] = s;
            }
        }
    }
    for (i = 0; i < v.size(); i++) cout << v[i] << " ";
    cout << endl;
}
```

Part 1

Given the following files:

```
UNIX> cat I1.txt
there's a port
UNIX> cat I2.txt
there's a port
on a western bay
UNIX>
```

Tell me the output of the following commands:

- **A:** program a < I1.txt
- **B:** program m < I1.txt
- **C:** program s < I1.txt
- **D:** program a < I2.txt
- **E:** program m < I2.txt
- **F:** program s < I2.txt

Part 2

What is the worst-case running time of this program if there are n words in standard input?

Part 3

Suppose you had to rewrite the program so that it runs more efficiently by using a more appropriate data structure from the and perhaps structuring the program differently. All that would matter is that the output is the same as **program.cpp**. Describe for me in a few sentences how you would do so. Make your description precise, and if you'd rather write code instead of a description, go ahead.

Question 6

Use the answer sheet for this question.

Below are six outputs of sorting programs. Each prints the original vector on the first line, and then state of the vector at each pass through the algorithm. The inputs are random doubles. For each output, state whether the sorting program is:

a. Bubble Sort.

b. Selection Sort.

c. Insertion Sort.

d. None of the above.

<p>Output 1:</p> <p>0.72 0.92 0.26 0.73 0.11 0.81 0.11 0.92 0.26 0.73 0.72 0.81 0.11 0.26 0.92 0.73 0.72 0.81 0.11 0.26 0.72 0.73 0.92 0.81 0.11 0.26 0.72 0.73 0.92 0.81 0.11 0.26 0.72 0.73 0.81 0.92</p>	<p>Output 2:</p> <p>0.95 0.15 0.69 0.52 0.04 0.11 0.15 0.95 0.69 0.52 0.04 0.11 0.15 0.69 0.95 0.52 0.04 0.11 0.15 0.52 0.69 0.95 0.04 0.11 0.04 0.15 0.52 0.69 0.95 0.11 0.04 0.11 0.15 0.52 0.69 0.95</p>
<p>Output 3:</p> <p>0.11 0.92 0.47 0.98 0.75 0.60 0.11 0.92 0.47 0.98 0.75 0.60 0.11 0.47 0.92 0.98 0.75 0.60 0.11 0.47 0.92 0.98 0.75 0.60 0.11 0.47 0.75 0.92 0.98 0.60 0.11 0.47 0.60 0.75 0.92 0.98</p>	<p>Output 4:</p> <p>0.81 0.31 0.21 0.02 0.05 0.31 0.21 0.31 0.81 0.02 0.05 0.31 0.31 0.81 0.21 0.02 0.05 0.31 0.02 0.05 0.21 0.31 0.81 0.31 0.02 0.21 0.31 0.81 0.05 0.31 0.02 0.05 0.21 0.31 0.31 0.81</p>
<p>Output 5:</p> <p>0.98 0.51 0.78 0.80 0.14 0.38 0.51 0.78 0.80 0.14 0.38 0.98 0.51 0.78 0.14 0.38 0.80 0.98 0.51 0.14 0.38 0.78 0.80 0.98 0.14 0.38 0.51 0.78 0.80 0.98 0.14 0.38 0.51 0.78 0.80 0.98 0.14 0.38 0.51 0.78 0.80 0.98</p>	<p>Output 6:</p> <p>0.85 0.22 0.52 0.26 0.12 0.60 0.12 0.22 0.52 0.26 0.85 0.60 0.12 0.22 0.52 0.26 0.85 0.60 0.12 0.22 0.26 0.52 0.85 0.60 0.12 0.22 0.26 0.52 0.85 0.60 0.12 0.22 0.26 0.52 0.60 0.85</p>

Extra Credit

You knew it was coming -- what song has **I2.txt** as its opening lyrics.