CSE1502 Fall 2007 Final Exam. Open book, open notes, no computers. Name ________________________

1. (2 points each) Given this declaration, what does the following print?

```
int i=0, a[]={10,20,30,40}, *p=a+1;
                                          ANSWERS
cout << 7/4 + 0.1;                            1.1
cout << string(5,'a').substr(1,3);            aaa
cout << atoi("12") + atoi("3");               15
cout << string("a") + toupper('a'+1);         aB
if (i++) cout << i; else cout << -i;          -1

cout << p - a;                                1
cout << *p - *a;                              10
cout << p[1];                                 30
cout << *(p+2) - 2;                           38
cout << ++*--p;                               11
```

2. (20 points) Write a function **isReverse** taking two **string**s and returning true if one is the reverse of the other, and false otherwise, e.g.

`isReverse("live", "evil")` returns true
`isReverse("aaa", "aa")` returns false

```
// ANSWER 1
bool isReverse(string a, string b)
{
  reverse(a.begin(), a.end());
  return a == b;
}

// ANSWER 2
bool isReverse(string a, string b)
{
  if (int(a.size()) != int(b.size()))
    return false;
  for (int i=0; i<int(a.size()); ++i)
    if (a[i] != b[int(a.size())-i-1])
      return false;
  return true;
}
```

3. (20 points) Write a function **longest** taking a vector<string> by reference and returns the longest string in the vector. In case of a tie, return the string that occurs first in lexicographical (ASCII code) order. If the vector is empty, return "", e.g.

```
  vector<string> v;
  cout << longest(v);  // prints nothing
  v.push_back("dog");
  v.push_back("cat");
  cout << longest(v);  // cat
  v.push_back("horse");
  cout << longest(v);  // horse
```

```
// ANSWER
string longest(vector<string>& v)
{
  string r = "";  // result
  for (int i=0; i<int(v.size()); ++i)
  {
    if (v[i].size() > r.size()
        || (v[i].size() == r.size()
            && v[i] < r))
      r = v[i];
  }
  return r;
}
```

4. (25 points) Write a program that takes a file name as a command line argument and prints the number of letters, lines and spaces in the file. Check errors as appropriate. For example, if **foo.txt** contains

```
This is just...
a
test!!!

a foo.txt
foo.txt has 15 letters, 3 lines, 2 spaces
a bar.txt
bar.txt not found
a
File name expected
```

```
// ANSWER
#include <iostream>
#include <fstream>
#include <cctype>
using namespace std;

int main(int argc, char **argv)
{
  if (argc < 2)
  {
    cout << "File name expected\n";
    return 0;
  }
  ifstream in(argv[1]);
  if (!in)
  {
    cout << argv[1] << " not found\n";
    return 0;
  }
  char c;
  int letters = 0, lines = 0, spaces = 0;
  while (in.get(c))
  {
    if (isalpha(c))
      ++letters;
    else if (c == '\n')
      ++lines;
    else if (c == ' ')
      ++spaces;
  }
  cout << argv[1] << " has "
       << letters << " letters, "
       << lines << " lines, "
       << spaces << " spaces\n";
  return 0;
}
```

5. (15 points) Class **Students** contains student grades stored in a map indexed by name. Write the non-inlined code for **getGrade**, which returns a student's grade given the name.

```
class Students
{
private:
  map<string, double> grades;
public:
  void setGrade(string name,double grade)
  {
    grades[name] = grade;
  }
  double getGrade(string name); // to do
};
```

```
// ANSWER
double Students::getGrade(string name)
{
  return grades[name];
}
```