CSE2050 Fall 2007 Exam #3. Open book, open notes, no computers. Name _____

1. What does the following print? (2 pts each)

```
char *a[3] = {"yes", "no", "maybe"};
```

	ANSWERS
`cout << a[1];`	**no**
`cout << a[1][1];`	**o**
`cout << a[2]+2;`	**ybe**
`cout << char(a[2][2]-1);`	**x**
`cout << *a;`	**yes**
`cout << **a;`	**y**
`cout << *a[1];`	**n**
`cout << (*a)+1;`	**es**
`cout << *(a+1);`	**no**
`cout << *(*a+1);`	**e**

2. Write a program that takes 2 command line arguments: a file name and a number n. The output is all the words in the file of length n, each on a separate line. A word is defined as a sequence of characters separated by white space. The program should give no output if the command line does not have exactly 2 arguments, or if the file does not exist. For example, if the file **in.txt** contains "this is a test", then: (40 pts)

```
a in.txt 4
this
test
```

```
// ANSWER
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;
int main(int argc, char **argv)
{
  if (argc != 3)
    return 0;
  ifstream in(argv[1]);
  if (!in)
    return 0;
  int n = atoi(argv[2]);
  string s;
  while (in >> s)
    if (int(s.size()) == n)
      cout << s << "\n";
  return 0;
}
```

3. Write a function **make_vector** taking a map<int,string> by reference and returning a vector<string> with the same contents. You may assume the keys of the map are nonnegative. The size of the vector should be the largest key of the map plus 1. Any elements with no corresponding keys should be assigned empty strings. For example:

```
map<int, string> m;
m[3] = "cat";
m[1] = "dog";
vector<string> v = make_vector(m);
```

would assign v a size of 4 where v[3] is "cat", v[1] is "dog", and v[0] and v[2] are "". (40 pts).

```
// ANSWER 1
vector<string> make_vector(
  map<int, string>& m)
{
  vector<string> v;
  for (map<int, string>::iterator
       p = m.begin(); p != m.end(); ++p)
  {
    while (int(v.size()) < p->first)
      v.push_back("");
    v.push_back(p->second);
  }
  return v;
}
```

```
// ANSWER 2
vector<string> make_vector(
  map<int, string>& m)
{
  vector<string> v;
  map<int, string>::iterator p = m.end();
  if (p != m.begin())
  {
    --p;
    v.resize(p->first + 1);
  }
  for (p = m.begin(); p != m.end(); ++p)
    v[p->first] = p->second;
  return v;
}
```