

1. What does the following print? (5 points each).

```
string s = "table";           ANSWERS  
cout << s[1];                a  
  
cout << s.substr(1, 3);      abl  
  
cout << s + s;               tabletable  
  
cout << "s" + s + 's';      stables  
  
cout << s[int(s.size()) - 2]; l
```

2. Write a function **average** that takes 4 arguments of type **double** and returns their average. The function should not print anything. (20 points).

```
// ANSWER  
double average(double a, double b,  
                double c, double d)  
{  
    return (a + b + c + d) / 4;  
}
```

3. Write a function **first** that takes 2 strings as arguments and returns whichever is shorter. If they are the same length, return whichever comes first in lexicographical order (i.e. the smaller when compared with the < operator). For example, **first("dog", "cat")** would return **"cat"**, and **first("apple", "fish")** would return **"fish"**. The function should not print anything. (25 points).

```
// ANSWER  
string first(string a, string b)  
{  
    int asize=int(a.size());  
    int bsize=int(b.size());  
    if (asize < bsize || (asize == bsize && a < b))  
        return a;  
    else  
        return b;  
}
```

4. Write a function **trim** that takes a string as an argument and removes any leading and trailing spaces. For example, if **s** is a string with the value "hello world " then **trim(s)**; would change the value of **s** to "hello world". If **s** is empty ("") or has no leading or trailing spaces, then it is unchanged. The function should not print or return anything. (30 points).

```
// ANSWER  
void trim(string& s)  
{  
    // Remove leading spaces  
    while (s != "" && s[0] == ' ')  
        s = s.substr(1);  
  
    // Remove trailing spaces  
    while (s != "" && s[int(s.size()) - 1] == ' ')  
        s = s.substr(0, int(s.size())-1);  
}
```