Test 1 sample question:

1. Write a **complete C++ program** for the following problem.

Prompt the user to enter the radius of a sphere. Then compute and display the surface area and volume of the sphere. The formulae for the computation are:

$$surface\ area = 4\pi R^2, \qquad volume = \frac{4}{3}\pi R^3$$

where R is the radius of the sphere.

You are required to write the following three user defined functions for this program:

- **GetRadius**: This function prompts the user to enter the radius value. It checks to see if the radius value is valid, i.e., it is greater than 0. It continues to prompt the user until a valid radius value is entered. It then returns the radius value to the calling function.
- **ComputeStatistics:** This function computes the **surface area** and the **volume** based on the radius of a sphere. The computed values are sent back to the calling function with reference parameters.
- **DisplayResults:** This function displays the radius value, the surface area, and the volume of a sphere.

Make sure to include all the function declarations and definitions. Think carefully whether to use void function or value returning function, and for parameter passing, whether to use value parameter or reference parameter.

2. Write a **C++ program** to count and display the number of occurrence of the word "**to**" in a line of text entered by user.

Example Run:

Enter Text: ***to be or not to be, that is going to be the hard question.*** ← *user input*

The word appeared 3 times in this sentence. ← *program output*

3. Write a complete C++ program to read a list of movie titles from a data file named "**movies.dat**". The movie titles are written one title per line in the data file. Your program reads the movie titles and stores them in an array named "**titles**". The capacity of the array is set to 100. The program reads the movie titles til the end of the data file is reached or a maximum of 100 values is read**.** The program then displays a message showing the number of movie titles read from the data file.

4. This problem has two parts.

(a) The function **Delete** does not work quite the way it is supposed to. Step through the code shown below and write down the output of the current version of the program.

```
#include <iostream>
using namespace std;
void Delete(int array[],  int position, int &size);
const  int MAX_SIZE=10;
int main()
{
     int values[MAX_SIZE]={4, 16, 3, 8, 20};
    int  size=5;
    int  k;
    int position = 1;

     // Delete the item at position in the array values
     Delete(values, position, size);

     for (k=0; k<size; k++)
        cout << values[k] << " ";
     cout << endl;
```

Your answer:

```
    return 0;
}

// This function deletes the element at location “position” in array A.
// If the position is < 0 or greater than the size of the array, nothing will be done.
// Otherwise, the value is deleted, and all the values below will be shifted up one position.
// It returns the new array size.
void Delete(int array[],  int position, int &size)
{
    for (int k=size; k>=position; k--)
       array[k] = array[k-1];

    size--;
}
```

(b) Modify function **Delete** to correctly delete an element at position "position" from the array. For example, before deletion, **array** looks like this: 5 7 9 2
After the deletion of element at position 1, **array** looks like this: 5 9 2