3 Exercises

1. The following is a program skeleton:

```
#include <iostream>
#include <cstring>    //for strlen(), strcpy()

struct stringy{
    char * str;   // points to a string
    int ct;      // length of string(not counting '\0')
};

// prototypes fo set(),show() and show() go here

int main()
{
    stringy beany;
    char testing[] = "Reality isn't what it used to be.";

    set(beany,testing);   // first argument is a reference,
                // allocates space to hold copy of testing,
                // sets str member of beany to point to the
                // new block, copies testing to new block,
                // and sets ct member of beany
    show(beany);      //print member string once
    show(beany, 2);  //prints member string twice

    testing[0] = 'D';
    testing[1] = 'u';
    show(testing);       //prints testing string once
    show(testing, 3);   //prints test string thrice
    show("Done!");

    return 0;
}
```

Complete this skeleton by providing the described functions and prototypes. Note that there should be two **show()** functions, each using default arguments. Use **const** arguments when appropriate. Note that **set()** should use **new** to allocate sufficient space to hold the designated string.

A sample runs might look like this:

```
Reality isn't what it used to be.
Reality isn't what it used to be.
Reality isn't what it used to be.
Duality isn't what it used to be.
Duality isn't what it used to be.
Duality isn't what it used to be.
Duality isn't what it used to be.
Done!
```

2. Write a template function max5() that takes as its argument an array of five items of type T and returns the largest item in the array. Test it in a program that uses the function with an array of five int values({1, 2, 3, 4 ,5}) and an array of five double values{1.1, 2.0, 3.0, 4.0, 5.5}.

A sample runs might look like this:

```
Max int = 5
Max double = 5.5
```

```
template <typename T>
T Max(T x, T y)
{
    return (x > y? x : y);
}
```