Lab12 myStrLen, myStrCpy

Work *with* a partner. Using the following prototypes

```
short myStrLen( char s[] );

short myStrLenP( char *ps );

void myStrCpy( char to[], char from[] );

void myStrCpyP( char *pTo, char *pFrom );
```

Write the following functions and a `main()` test suite that verifies that the functions work correctly.

(0) Write `myStrLen` that uses character array `[ ]` arguments.

(1) Write `myStrLenP` that uses character pointer arguments.

(2) Write `myStrCpy` that uses character array `[ ]` arguments.

(3) Write `myStrCpyP` that uses character array arguments.

Note: The character array and pointer versions can be *called* (from main()) the same way, that is:

```
char s[20];
strcpy(s, “foobar”); // yes, you can use strcpy() to setup

short Length;

Length = myStrLen( s );
cout << “Length of “ << s << “ is: “ << Length;

Length = myStrLenP( s );
cout << “Length of “ << s << “ is: “ << Length;
```