Assignment 5

In this assignment you will create some functions to work with strings. You will get some practice with strings, pointers, and structures.

Problem 1.

Create the functions described below. use the provided `main` function to test them.

Function 1: Lower-case a string

Write a function that will lower-case a string passed as a parameter to the function. Use the following template for the function:

```
void strlower(char* instring)
{
   ... your code here
}
```

There should be no reason to allocate memory for this function. you can use the lower() function from earlier lectures to make this function work.

Function 2: Character Search

Write the following function that will search for a character in a string. again, use the template below:

```
// search for character c in instring
// if found, return the index in the string, otherwise return -1
char stringchar(char* instring, char c)
{

}
```

Test these functions with the following code:

```
#include <stdio.h>
#include <stdlib>
int main(void)
{
    char* str1 = "Hello World";
    strlower(str1);
    printf("%s\n", str1);

    printf("%c should be 0\n", stringchar(str1, 'h'));
    printf("%c should be 4\n", stringchar(str1, 'o'));
    printf("%c should be -1\n", stringchar(str1, 'x'));

    return EXIT_SUCCESS;
}
```

Problem 2

Write a program that uses the following structure for holding a string and it’s length. This is similar to what many languages do for keeping strings in memory.

```
typedef struct {
    char* stringtext;  // a pointer to the string
    unsigned int length; // the length of the string
} String;
```

Write a constructor that allocates memory to a string object dynamically using malloc and then initializes the object with data passed in. your constructor should have the following signature:

```
String* makeString(char* instring)
{
   ... your code here.
}
```

Use the `makestring` function from the example code to allocate the `stringtext` property in the structure.

Remember that `structs` are like classes with no methods. all the ‘methods’ are standalone.

use the following code to test this string.

```
#include <stdio.h>
#include <stdlib>
int main(void)
{
    char* str1 = "Hello World";
    String* hello = makeString(str1);

    printf("the String hello contains %s which is %u in length\n", hello->stringtext, hello->length);
}
```

Finally, free all the data you dynamically allocated before the program ends.

To turn this assignment in, zip up you source code and submit to canvas.