Echahid Hamma Lakhdar University of El-Oued
Department of Computer Science
Level: 2nd Year LMD Computer Science
Course: Algorithms and Data Structures
Lab Work No. 2

(Strings, Pointers, Dynamic Memory, functions and Recursion)

We aim to write in the C language a library of independent subprograms that provide several functionalities on strings represented by different data structures — mainly arrays, and sometimes pointers.
This lab work has the following objectives:
- To become familiar with pointers and dynamic memory allocation.
- To write a set of C functions that can form a library, designed to be independent of any program that may use them.

Main Program:

```
#include< stdio.h >
#include< string.h >
#include< stdlib.h >
char *LoadString(int N);
int StringLength(char *str);
void LoadArray(char *p, char arr[]);
void ReverseArray(char arr[], char rev[], int n);
void DisplayArray(char arr[], int n);
int SumStringASCII(char *p);
void ReverseString(char *start, char *end);
int main() {
char *str;
int n;

printf("Please enter the maximum size of the string:\n");
scanf("%d", &n);
getchar();
str = LoadString(n);
int len = StringLength(str);
char arr[len + 1], rev[len + 1];
LoadArray(str, arr);
printf("\n Original array: ");
DisplayArray(arr, len);
ReverseArray(arr, rev, len);
printf("\n Reversed array: ");
DisplayArray(rev, len);
int sum = SumStringASCII(str);
printf("\n Sum of ASCII values (recursive): ReverseString(str, str + len
- 1);
printf("String reversed recursively: %s\n", str);
free(str);
return 0;
}
```

- Write a function `char *LoadString(int N);` that allows reading a string of characters whose size `N` is entered by the user. This function allocates memory dynamically using `malloc()` and returns a pointer to the string read.
- Write a function `int StringLength(char *str);` that computes and returns the length of the string entered by the user (without using `strlen()`).
- Write a procedure `void LoadArray(char *p, char arr[]);` that loads the string pointed to by `p` into a character array `arr`.
- Write a procedure `void ReverseArray(char arr[], char rev[], int`

`n);` that reverses the array `arr` and stores the result in another array `rev` of the same type and length.

- Write a procedure `void DisplayArray(char arr[], int n);` that displays a character array `arr` as a string using the length `n` as a parameter.
- Write a recursive function `int SumStringASCII(char *p);` that computes and returns the sum of the ASCII codes of all characters in the string pointed to by `p`. The function should stop when the null character ‘”‘ is reached.
- Write a recursive function `void ReverseString(char *start, char *end);` that reverses a string in place using pointers to the first and last characters (`start` and `end`). The function should recursively swap characters until all are reversed.
- Use the `main()` function provided to test all of the above functionalities.