BIL105E – Introduction to Scientific and Engineering Computing (C)

Spring 2015 - 2016

CRN: 21837

Homework 3

Assignment Subject: ***Writing a C program which the user implements some character sequence operations by using pointers and functions.***

Assignment Due Date: ***01.05.2016 23:00***

Submission Date: ***01.05.2016***

Student Name: ***Korel Chairoula***

Student ID: ***150140910***

e-mail: ***korel_hayrullah@hotmail.com***

Introduction Section

First of all, this project was a little bit challenging comparing to the other projects. This project is all about string manipulation. The user enters a string and after that it processes the string and return it back to the user. In this project there were 6 steps to implement. At the same time, all the variables used in this project needed dynamic memory allocation and reallocation. Of course after using the allocated memory the allocated memory should be freed in order to use it again. At the very beginning, there comes a menu for the user to choose what to do. 0 is for exiting the program. 1 is for setting the current character sequence which must be set at first to continue on the other steps. If the user enters the other options except 0 it warns the user that he/she has to set the current character sequence. After setting the current character sequence, there are other 4 options which are: 2 for Getting Substring which the user enters the gap and prints the selected gap on screen. If the gap does not fit in the entered current character sequence user gets a warning that it is NULL. 3 for Removing Substring which like the same with Get Substring the user enters a gap and this function takes the gap and removes the selected gap. If the gap is not correct and does not match with the Current Character Sequence also it warns the user that it is NULL. 4 is for Inserting Substring which at first gets the string that will inserted and after getting the string it takes a beginning index in which the program starts inserting the entered string just after the begin index. For example, let's say our current character sequence is RABBIT. We want to add F with starting index 2. Keep in mind that is starts counting from 0. So, the result will be RAFBBIT. Lastly, 5 is for Replace which replaces any other entered string from the user with the original one. First, the after entering 5 as option the program asks from the user to enter a string the to be found. After that, the program asks from the user to enter a string which will be replaced with found string or character. For instance, our current character sequence is PLAY. We will replace TWO with A. The final result will be PLTWOY. However, if the string or character entered to be found is not matching, it will not make any change. The goal of this project is to understand how strings are working on programming languages like C, in which strings are always confusing and driving the coder crazy.

Development Environment

Operating System: ***OSX El Capitan***
Programming Language: ***C***
Compiler: ***GCC***
Source File*: My program includes one file c named* ***150140910.c***
Header Files: *The header files used are* ***<stdio.h>, <stdlib.h> and <string.h>.***

I used header files <stdio.h> because it is the default library in order to run default functions. After that, I used <stdlib.h> for malloc(), realloc() and free() functions in order to allocate, reallocate memory and freeing the allocated memory from the variables. Finally, I used <string.h> library for some string functions which in my project I used strcpy() and strcat(). Strcpy() function stands for "String Copy" and strcat() function for "string concatenation". At the very beginning, I defined the MAX_CHARACTER to 81 to use it in my

program when needed. For example, “define MAX_CHARACTER 81”, so whenever I needed it I wrote only the name of it. After finishing coding my program I used terminal on my MacBook to compile it via gcc. My command was “Korel-MacBook-Pro:~ Korel.Hayrullah$ gcc 150140910.c”. In order to run it I used in terminal “./a.out” to run my program. There is a easier way to compile and simultaneously change the name is “gcc 150140910.c –o MyThirdAssignmentC” and then again to run it “./MyThirdAssignmentC.

Data Structure and Variables

Nearly all the explanations for the variables and information for the code’s data structure and variables are explained in the code.

Function Prototypes

```
int user_menu(void);
int set_ccs(char **ccs);
char *sub_string (char *ccs, int begin_index, int end_index);
char *remove_string(char **ccs, int begin_index, int end_index);
int insert_string(char **ccs, char *insert, int begin_index);
int replace_string(char **ccs, char *find, char *replace);
```

Variables defined in main

```
char **ccs;

char *insert;

char *find;

char *replace;

char *temporary;

char *temporary2;

int temporary3;

int temporary4;

int begin_index;

int end_index;
```

int control;

Variables defined in user_menu function

int control;

Variables defined in set_ccs function

int length;

Variables defined in sub_string function

char *temporary;

int allocation;

int l;

int j = 0;

Variables defined in remove_string function

char *temporary2;

char *hold;

int allocation;

int length;

int reallocation;

int i;

int j = 0;

Variables defined in insert_string function

int length;

int length2;

int length3;

```
int I;
int j = 0;
```

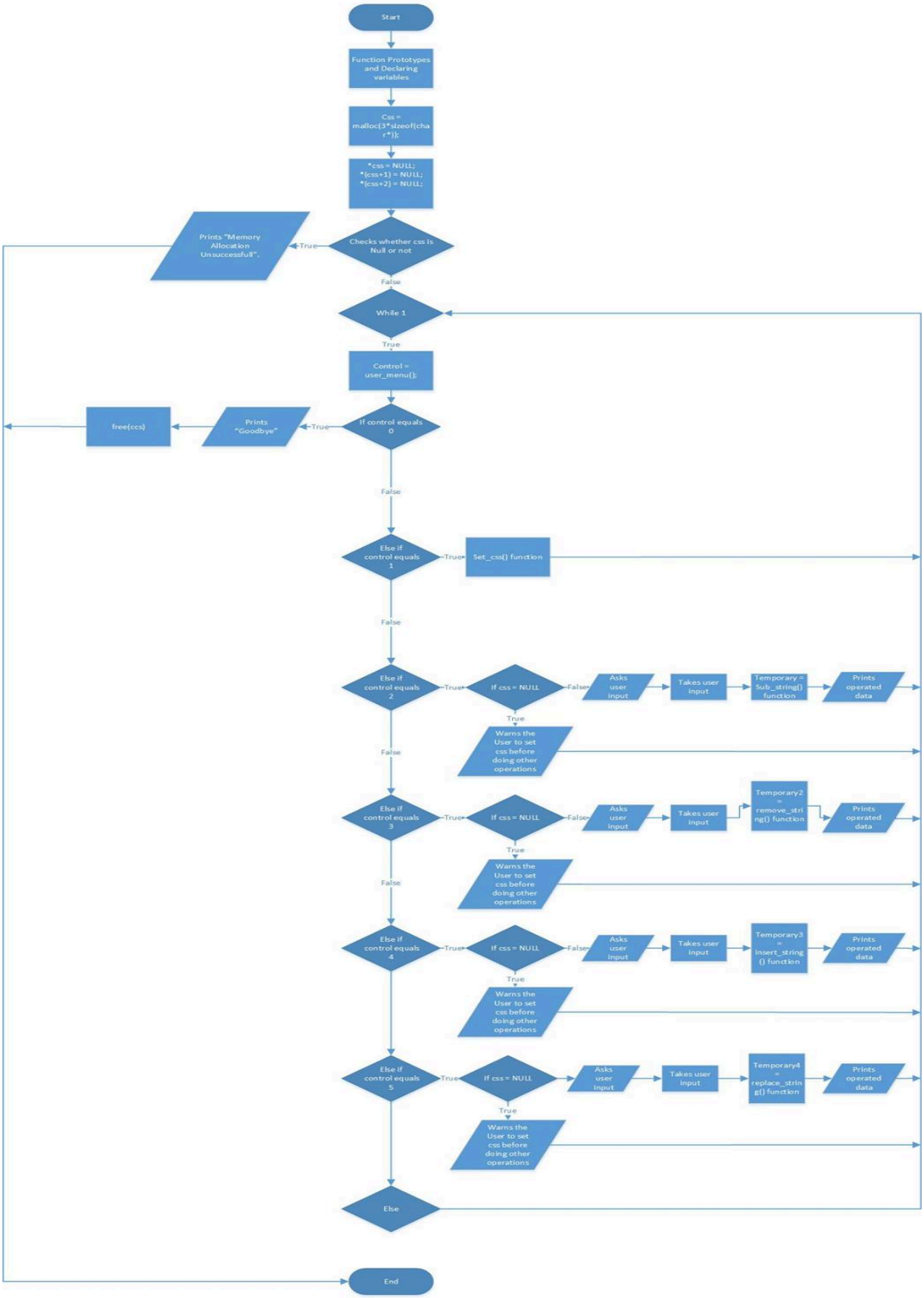
Variables defined in replace_string function

```
int counter = 0;

int i = 0;

int j;

int begin_index;

int end_index;
```

Program Flow

Flowchart of the Program

Main Function



Conclusion

While working on this project, I came across with a lot of problems. Sometimes, it took me hours to understand the fault because sometimes it showed me the error and sometimes not. This project was not simple and it was confusing a lot. However, at the end of this project I understood the principle of pointers and strings which in my opinion it is the most difficult parts to understand how it works in programming languages like C. All the time, I had to figure out the problem on paper. Also, in this project I understood how to allocate memory and reallocate memory which is dynamic memory allocation. After using memory allocation at the end when it is not needed anymore you have to free the used memory. All in all, this project was really challenging and took a lot of time to understand the faults I got and at the end gave me a lot of knowledge and experience about strings and pointers.