Week 5

ANIRUDH ROWJEE `PES2UG20CS050`

Task 1

Aim: Write functions to

- Reverse a String
- Check for the Equality of two strings

`main.c`

```c
#include <stdio.h>
#include <stdbool.h>
#include <strings.h>
#include "utils.h"


/*
 * Aim: Write functions to
 * a) Reverse a String
 * b) Check for the Equality of two strings
 */

int main()
{
    // declare a blank string, we will fix length overflows using putchar
    char input[500];
    int string_length = 0;

    printf("Enter the string > ");
    char temp;
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        input[string_length] = temp;
        string_length++;
    }

    // declare the array for the reversed string to go in
    char output[string_length+1];
    output[string_length] = '\0';

    reverse_string(input, output, string_length);
    printf("the reversed string is %s\n", output);

    bool status = strings_are_equal(input, output, string_length);
    printf("The Strings are %s\n", status ? "Equal" : "Not Equal");
```

```
        return 0;
    }
```

utils.h

```c
void reverse_string(char* input_string, char* reversed_string, int string_length)
bool strings_are_equal(char* first, char* second, int string_length);
```

utils.c

```c
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include "utils.h"

void reverse_string(char* input_string, char* reversed_string, int string_length)
{
    // index of the final character in the input array
    int final_character = string_length-1;
    int offset = 0;
    for (int i = 0; i < string_length; i++)
    {
        reversed_string[i] = input_string[final_character - offset];
        offset++;
    }
}

bool strings_are_equal(char* first, char* second, int string_length)
{
    // function to check if two strings are equal
    bool equal = true;

    for (int i = 0; i < string_length; i++)
    {
        if (first[i] != second[i])
        {
            equal = false;
        }
        return equal;
    }

}
```

Output

```
nidavellir :: UE20CS152/week4/t1 » ./main
Enter the string > hello
```

```
the reversed string is olleh
The Strings are Not Equal
nidavellir :: UE20CS152/week4/t1 » ./main
Enter the string > malayalam
the reversed string is malayalam
The Strings are Equal
nidavellir :: UE20CS152/week4/t1 » ./main
Enter the string > racecar
the reversed string is racecar
The Strings are Equal
nidavellir :: UE20CS152/week4/t1 »
```

Task 2

write a function to find all the occurrances of a character in a string, and use this function to replace all said occurrances by another character.

`main.c`

```c
#include <stdio.h>
#include <stdbool.h>
#include "utils.h"

/* write a function to find all the occurrances of a character
 * in a string, and use this function to replace all said
 * occurrances by another character
 */

int main()
{
    // declare the string we'll be using
    char input[500];
    int string_length = 0;
    char temp;
    printf("Enter the string > ");

    while ((temp = getchar()) != EOF && temp != '\n')
    {
        input[string_length] = temp;
        string_length++;
    }

    char find, replace;
    printf("Enter the Characters to look for and replace > ");
    scanf("%c %c", &find, &replace);
    find_and_replace(input, string_length, find, replace);

    printf("The substituted string is > ");
    for (int i = 0; i < string_length; i++)
    {
        printf("%c", input[i]);
```

```c
    }
    printf("\n");

    return 0;
}
```

utils.h

```c
void find_and_replace(char* input_string, int string_length,
        char find, char replace);
```

utils.c

```c
#include <stdio.h>


void find_and_replace(char* input_string, int string_length,
        char find, char replace)
{
    for (int i = 0; i < string_length; i++)
    {
        if (input_string[i] == find)
        {
            input_string[i] = replace;
        }
    }
}
```

Output

```
nidavellir :: UE20CS152/week4/t2 » ./main
Enter the string > hi this is anirudh
Enter the Characters to look for and replace > i @
The substituted string is > h@ th@s @s an@rudh
nidavellir :: UE20CS152/week4/t2 » ./main
Enter the string >
Enter the Characters to look for and replace >  $
The substituted string is > $$$
```

Task 3

Write a function to remove all repeated characters from a given string and display the string without duplicate characters

main.c

```c
#include <stdio.h>
#include "utils.h"

/*
 * Write a function to remove all repeated characters from
 * a given string and display the string without duplicate
 * characters
 *
 */

int main()
{
    char input[500];
    int string_length = 0;
    char temp;

    printf("Enter the string > ");
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        input[string_length] = temp;
        string_length++;
    }
    printf("The string without any repeated characters > ");
    displayWithoutRepeatedCharacters(input, string_length);
    return 0;
}
```

utils.h

```c
void displayWithoutRepeatedCharacters(char* input, int strlen);
```

utils.c

```c
#include <stdio.h>
#include "utils.h"

void displayWithoutRepeatedCharacters(char* input, int strlen)
{
    // rudimentary hashtable
    // supports ASCII only
    int hashtable[127];
    for (int i = 0; i < 127; i++)
        hashtable[i] = 0;

    // create the new string, and fill it with spaces
    // this is a new string with the maximum length
    // of the old string, given that you cannot have more
    // unique characters than total characters in a string
    char new[strlen];
```

```c
    int new_length = 0;

    for (int x = 0; x < strlen; x++)
    {
        new[x] = ' ';
    }

    for (int i = 0; i < strlen; i++)
    {
        char current = input[i];

        // get ASCII value of character
        int hash = (int)current;

        // check if the character's a repeated one
        if (hashtable[hash] == 0)
        {
            new[new_length] = current;
            new_length++;
        }
        hashtable[hash]++;
    }
    printf("%s\n", new);

}
```

Output

```
nidavellir :: UE20CS152/week4/t3 » ./main
Enter the string > pani puri
The string without any repeated characters > pani ur
nidavellir :: UE20CS152/week4/t3 » ./main
Enter the string > interesting
The string without any repeated characters > intersg
```

Task 4

Take two strings and append the second one to the first one `n` times.

`main.c`

```c
#include <stdio.h>
#include "utils.h"

int main()
{

    char string1[500], string2[500];
    int length1 = 0, length2 = 0, repeat = 0;
```

```c
    char temp;

    printf("Enter the first string > ");
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        string1[length1] = temp;
        length1++;
    }

    printf("Enter the second string > ");
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        string2[length2] = temp;
        length2++;
    }

    printf("Enter the number of repeats > ");
    scanf("%d", &repeat);

    concatenateNTimes(string1, string2, length1, length2, repeat);

    return 0;
}
```

utils.h

```c
void concatenateNTimes(char* input1, char* input2, int l1, int l2, int n);
```

utils.c

```c
#include <stdio.h>

void concatenateNTimes(char* input1, char* input2, int l1, int l2, int n)
{
    // printf("%s %d %s %d %d\n", input1, l1, input2, l2, n);
    int new_length = l1 + (n * l2);
    printf("The new length is %d\n", new_length);

    char new_string[new_length];
    for (int i = 0; i < new_length; i++)
    {
        new_string[i] = '*';
    }

    for (int i = 0; i < l1; i++)
    {
        new_string[i] = input1[i];
    }
```

```c
    // printf("%s\n", new_string);

    for (int i = 0; i < n; i++)
    {
        int offset = l1 + (i * l2);
        for (int j = 0; j < l2; j++)
        {
            new_string[offset + j] = input2[j];
        }
        // printf("%s\n", new_string);
    }
    printf("The repeated string is > ");
    printf("%s\n", new_string);
}
```

Output

```
nidavellir :: UE20CS152/week4/t4 » ./main
Enter the first string > try
Enter the second string > hard
Enter the number of repeats > 3
The new length is 15
The repeated string is > tryhardhardhard
nidavellir :: UE20CS152/week4/t4 » ./main
Enter the first string > nice
Enter the second string > day
Enter the number of repeats > 5
The new length is 19
The repeated string is > nicedaydaydaydayday
```

Practice Program 1

Find the number of occurrances of each character in a string.

`main.c`

```c
#include <stdio.h>
#include "utils.h"

int main()
{
    char new_string[500];
    int string_length = 0;
    char temp;

    printf("Enter the string to Analyze > ");
    while((temp = getchar()) != EOF && (temp != '\n'))
    {
        new_string[string_length] = temp;
```

```c
            string_length++;
    }

    char_frequency_distribution(new_string, string_length);

    return 0;
}
```

utils.h

```c
void char_frequency_distribution(char* input, int strlen);
```

utils.c

```c
#include <stdio.h>
#include "utils.h"


void char_frequency_distribution(char* input, int strlen)
{
    // rudimentary hashtable
    // supports ASCII only
    int hashtable[127];
    for (int i = 0; i < 127; i++)
        hashtable[i] = 0;

    for (int i = 0; i < strlen; i++)
    {
        char current = input[i];

        // get ASCII value of character
        int hash = (int)current;

        hashtable[hash]++;
    }

    for (int i = 0; i < 127; i++)
    {
        int current = hashtable[i];
        // display only those characters present
        if (current >= 1)
        {
            printf("Character %c appears %d times\n", i, current);
        }
    }
}
```

Output

```
nidavellir :: UE20CS152/week4/pp1 » ./main
Enter the string to Analyze > othello
Character e appears 1 times
Character h appears 1 times
Character l appears 2 times
Character o appears 2 times
Character t appears 1 times
nidavellir :: UE20CS152/week4/pp1 » ./main
Enter the string to Analyze > banana
Character a appears 3 times
Character b appears 1 times
Character n appears 2 times
nidavellir :: UE20CS152/week4/pp1 » ./main
Enter the string to Analyze > malayalam
Character a appears 4 times
Character l appears 2 times
Character m appears 2 times
Character y appears 1 times
```

Practice Program 2

Given two strings, check if the second one is present at the end of the first.

`main.c`

```c
#include <stdio.h>
#include <stdbool.h>
#include "utils.h"

int main()
{

    char string1[500], string2[500];
    int length1 = 0, length2 = 0;

    char temp;

    printf("Enter the first string > ");
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        string1[length1] = temp;
        length1++;
    }

    printf("Enter the second string > ");
    while ((temp = getchar()) != EOF && temp != '\n')
    {
        string2[length2] = temp;
        length2++;
    }
```

```c
    bool present = checkIfPresentAtEnd(string1, string2,
            length1, length2);
    printf("The second string %s at the end of the first\n",
            present ? "is present" : "is not present"
            );
    return 0;
}
```

utils.h

```c
bool checkIfPresentAtEnd(char* a, char* b, int l1, int l2);
```

utils.c

```c
#include <stdio.h>
#include <stdbool.h>
#include "utils.h"


bool checkIfPresentAtEnd(char* a, char* b, int l1, int l2)
{
    bool present = true;

    // assumes L1 is longer than l2
    int start = l1 - l2;

    for (int i = 0; i < l2; i++)
    {
        if (a[start + i] != b[i])
            present = false;
    }

    return present;
}
```

Output

```
nidavellir :: UE20CS152/week4/pp2 » ./main
Enter the first string > hello
Enter the second string > lo
The second string is present at the end of the first
nidavellir :: UE20CS152/week4/pp2 » ./main
Enter the first string > worldworldworld
Enter the second string > world
The second string is present at the end of the first
nidavellir :: UE20CS152/week4/pp2 » ./main
Enter the first string > absence
```

```
Enter the second string > enc
The second string is not present at the end of the first
```