Recep Tayyip Erdogan University

Faculty of Engineering and Architecture

Computer Engineering

CE103- Algorithms and Programming I

Homework-1 (Week-3)

Fall Semester, 2020-2021

Instructor	**Asst. Prof. Dr. Uğur CORUH**
Contact Information	ugur.coruh@erdogan.edu.tr
Google Classroom Code	**ouw44uk**
Publish Date	**28.10.2020**
Due Date	**06.11.2020 17:00**

Complete the following homework requirements, prepare them in the format given in the link below until the deadline and time, and upload them to the related assignment in the classroom.

https://drive.google.com/file/d/1yqSXZZ3346iIqotb_e_yzaryfxEXE0fR/view?usp=sharing

Grades:

Problem-1	20 points
Problem-2	15 points
Problem-3	15 points
Problem-4	15 points
Problem-5	15 points
Problem-6	20 points
Total	**100** points

You will develop the following examples in C language. The examples with C ++ will not be accepted.

NOTE: In the code, add the following information with printf. Put the description of each application in the problem part.

```
int main(void)
{
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Printing the Text Entered on the Screen in Reverse
\n");

... your codes...

}
```

Problem-1: *Printing the Text Entered on the Screen Reverse (20 points)*

When the application is opened, it waits for an input like "please enter the text:". After the entry is entered, "hello" is output as "olleh". Then it waits as "Do you want to enter a new entry (y / n)" y: yes n: no. When the y character is entered, it clears the screen and displays the text "please enter the text:" and waits for the content to be entered in the same way. Otherwise close application with “bye” message.

```
#include <stdio.h>  //for printf
#include <stdbool.h> //for bool

//fgets reads lines and this is the maximum length of read buffer. There must
be a limit
#define MAX_SENTENCE_LENGTH 500

//Reverse string with recursive operation.
void reverse(char *str);

//Entrypoint of Application
int main()
{

      //Create static buffer with length and initiate with zeros
   char buffer[MAX_SENTENCE_LENGTH] = {0};
   //Decleare a boolean variable flag for repeated readings
   bool again = true;
   //Decleare a command variable to store yes or no command.
   char command = NULL;

      //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Printing the Text Entered on the Screen in Reverse
\n");

   //First show message
   printf("please enter the text:\n");

   //Create a infinite loop for continues reading and exit from loop with
again flag
   while(true)
   {
            //this is important you should clear console keyboard input
buffer
            fflush(stdin);

            //read line and check also if it is not NULL
            if(fgets(buffer,MAX_SENTENCE_LENGTH-1,stdin)!=NULL)
            {
                  //call reverse function this will print reverse of entered
sentence
                  reverse(buffer);

                  //clear input buffer
```

```
                    memset(buffer,0,sizeof(buffer));

                    //write newline to seperate inputs.
                    printf("\n\n");

                    //this second loop is check for command inputs we only
accept Y/y or N/n
                    while(true){

                          //ask question
                          printf("Do you want to enter a new entry (y /
n):\n");

                          //clear console inputs we will ask for input
                        fflush(stdin);

                          //read single character
                        command = getchar();


                          if(command=='Y' || command == 'y'){
                                //if its YES
                                //print message and go up from loop to continue
for next reading...
                                printf("please enter the text:\n");
                                //exit from command control loop
                                break;
                          }else if(command=='N' || command == 'n'){
                              //if its NO
                              //printf BYE
                                printf("bye\n");
                                //set again flag false to break loop
                             again = false;
                             //exit from command control loop
                             break;
                          }else{
                                //printf warning message
                                printf("Please just enter Y/y for YES or N/n
for NO\n");
                                //continue from command check loop
                          }
                    }

                    //if NO selected then loop will exit from this point.
                    if(again==false)
                          break;
              }
    }

       //wait a input to close program
    getchar();

    return 0;
}

//if input is not NULL then push this call operations to call-stack
//and moves to next string by increment of adress and
//if adress is NULL then program terminated and recursively empty call-stack
//this will cause print stack printf operation from end to begining
```

```c
//and this will reverse all sentence or inputs
void reverse(char *str)
{
   if (*str)
   {
        reverse(str+1);
        printf("%c", *str);
   }
}
```

Problem-2: *Drawing a Triangle with the Star Character on the Screen*

When the application opens, it creates the following figure on the screen using loops.

```
*
**
***
*****
*****
****
***
**
*
```

```
#include<stdio.h>

int main()
{
    int i, j, N, columns;

      //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Drawing a Triangle with the Star Character on the
Screen \n");

    /* Input number of columns from user */
    printf("Enter number of columns:");
    scanf("%d",&N);

    columns=1;

    for(i=1;i<N*2;i++)
    {
        for(j=1; j<=columns; j++)
        {
            printf("*");
        }

        if(i < N)
        {
            /* Increment number of columns per row for upper part */
            columns++;
        }
        else
        {
            /* Decrement number of columns per row for lower part */
            columns--;
        }

        /* Move to next line */
        printf("\n");
    }
    return 0;}
```

Problem-3: *Fibonacci Sequence (15 points)*

Write the application that creates the fibonacci sequence as the number entered on the screen.

```c
#include <stdio.h>

int main()
{
    int a, b, c, i, terms;


      //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Fibonacci Sequence\n");

    /* Input number from user */
    printf("Enter number of terms: ");
    scanf("%d", &terms);

    /* Fibonacci magic initialization */
    a = 0;
    b = 1;
    c = 0;

    printf("Fibonacci terms: \n");

    /* Iterate through n terms */
    for(i=1; i<=terms; i++)
    {
        printf("%d, ", c);

        a = b;      // Copy n-1 to n-2
        b = c;      // Copy current to n-1
        c = a + b; // New term
    }

    return 0;
}
```

Problem-4: *Number Base Conversions (15 points)*

Write the code that writes the equivalents of all the numbers in the range of numbers on the screen in the binary, hexadecimal, octal system.

```
#include <stdio.h>

#define MAX_STRING_BUFFER 50

int main()
{

      char binaryString[MAX_STRING_BUFFER] = {0};
      char octalString[MAX_STRING_BUFFER]={0};
      char hexString[MAX_STRING_BUFFER]={0};

      int upperBound = -1;
      int lowerBound = -1;
      int i = 0;

      //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Number Base Conversions  \n");

    /* Input number from user */
    printf("Enter upper bound: ");
    scanf("%d", &upperBound);

    printf("Enter lower bound: ");
    scanf("%d", &lowerBound);

for (i=lowerBound; i<=upperBound;i++){

      printf("Number: [%d]\n",i);

      itoa(i,binaryString,2);
      printf("Binary=%s\n",binaryString) ;

      itoa(i,octalString,8);
      printf("Octal=%s\n",octalString) ;

      itoa(i,hexString,16);
      printf("Hexadecimal=%s\n",hexString) ;

      printf("**********************************\n");

}

    return 0;
}
```

Problem-5: *Prime Factorization (15 points)*

Write the code that divides the number it takes as input from the screen into prime factors.

```c
/**
 * C program to find all prime factors of a given number
 */

#include <stdio.h>

int main()
{
    int i, j, num, isPrime;

            //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "Number Base Conversions  \n");

    /* Input a number from user */
    printf("Enter any number to print Prime factors: ");
    scanf("%d", &num);

    printf("All Prime Factors of %d are: \n", num);

    /* Find all Prime factors */
    for(i=2; i<=num; i++)
    {
        /* Check 'i' for factor of num */
        if(num%i==0)
        {
            /* Check 'i' for Prime */
            isPrime = 1;
            for(j=2; j<=i/2; j++)
            {
                if(i%j==0)
                {
                    isPrime = 0;
                    break;
                }
            }

            /* If 'i' is Prime number and factor of num */
            if(isPrime==1)
            {
                printf("%d, ", i);
            }
        }
    }

    return 0;
}
```

Problem-6: *strcat, strcmp and strcpy function (20 points)*

Write the standard functions strcat used to add strings of characters, strcmp used to compare strings and strcpy used to copy strings.

```
#include <stdio.h>

#define MAX_BUFFER 50

char *my_strcat(char *strg1, char *strg2);
int my_strcmp(char *strg1, char *strg2);
char *my_strcpy(char *destination, char *source);

int main()
{
    char strg1[MAX_BUFFER] = {0};
    char strg2[MAX_BUFFER] = {0};

      //Your identification data...
      printf("Build Time: %s %s\n", __DATE__, __TIME__);
      printf("Owner: Name Surname\n");
      printf("Student ID: 11111111\n");
      printf("Course: CE-103\n");
      printf("Homework: 1\n");
      printf("Problem: "strcat, strcmp and strcpy function   \n");

      printf("\n...strcat demo...\n");

    printf("Enter first string: ");

    fflush(stdin);
    fgets(strg1,MAX_BUFFER-1,stdin);


    printf("Enter second string: ");

    fflush(stdin);
    fgets(strg2,MAX_BUFFER-1,stdin);

    printf("\nConcatenating first and second string .. \n\n");
    my_strcat(strg1, strg2);

    printf("First string: %s\n", strg1);
    printf("Second string: %s", strg2);

    printf("\n...strcmp demo...\n");

    printf("Enter first string: ");

    fflush(stdin);
    fgets(strg1,MAX_BUFFER-1,stdin);


    printf("Enter second string: ");

    fflush(stdin);
    fgets(strg2,MAX_BUFFER-1,stdin);

    if(my_strcmp(strg1, strg2)==0){
```

```
        printf("\nYou entered the same string two times");
    }else{
        printf("\nEntered strings are not same!");
    }

      printf("\n...strcpy demo...\n");

    printf("Enter first string: ");

    fflush(stdin);
    fgets(strg1,MAX_BUFFER-1,stdin);


    printf("Enter second string: ");

    fflush(stdin);
    fgets(strg2,MAX_BUFFER-1,stdin);

    printf("\nCopying first string into second... \n\n");
    my_strcpy(strg2, strg1); // copy the contents of strg1 to strg2

    printf("First string = %s\n", strg1);
    printf("Second string = %s\n", strg2);

    return 0;
}

char *my_strcat(char *strg1, char *strg2)
{
    char *start = strg1;

    while(*strg1 != '\0')
    {
        strg1++;
    }

    while(*strg2 != '\0')
    {
        *strg1 = *strg2;
        strg1++;
        strg2++;
    }

    *strg1 = '\0';
    return start;
}

int my_strcmp(char *strg1, char *strg2)
{
    while( ( *strg1 != '\0' && *strg2 != '\0' ) && *strg1 == *strg2 )
    {
        strg1++;
        strg2++;
    }

    if(*strg1 == *strg2)
    {
        return 0; // strings are identical
    }
```

```
    else
    {
        return *strg1 - *strg2;
    }
}

char *my_strcpy(char *destination, char *source)
{
    char *start = destination;

    while(*source != '\0')
    {
        *destination = *source;
        destination++;
        source++;
    }

    *destination = '\0';
    return start;
}
```