

MCS 011 Problem Solving and Programming Solved Assignment

Q1. Write an algorithm, draw corresponding flow chart and write an interactive program which prompts the user with the following options on the opening menu:

- 1. To accept two integers and check whether they are equal or not**
- 2. To check whether a given number is even or odd**
- 3. To check whether a given number is a positive number or a negative number**
- 4. Quit**

Enter your choice:

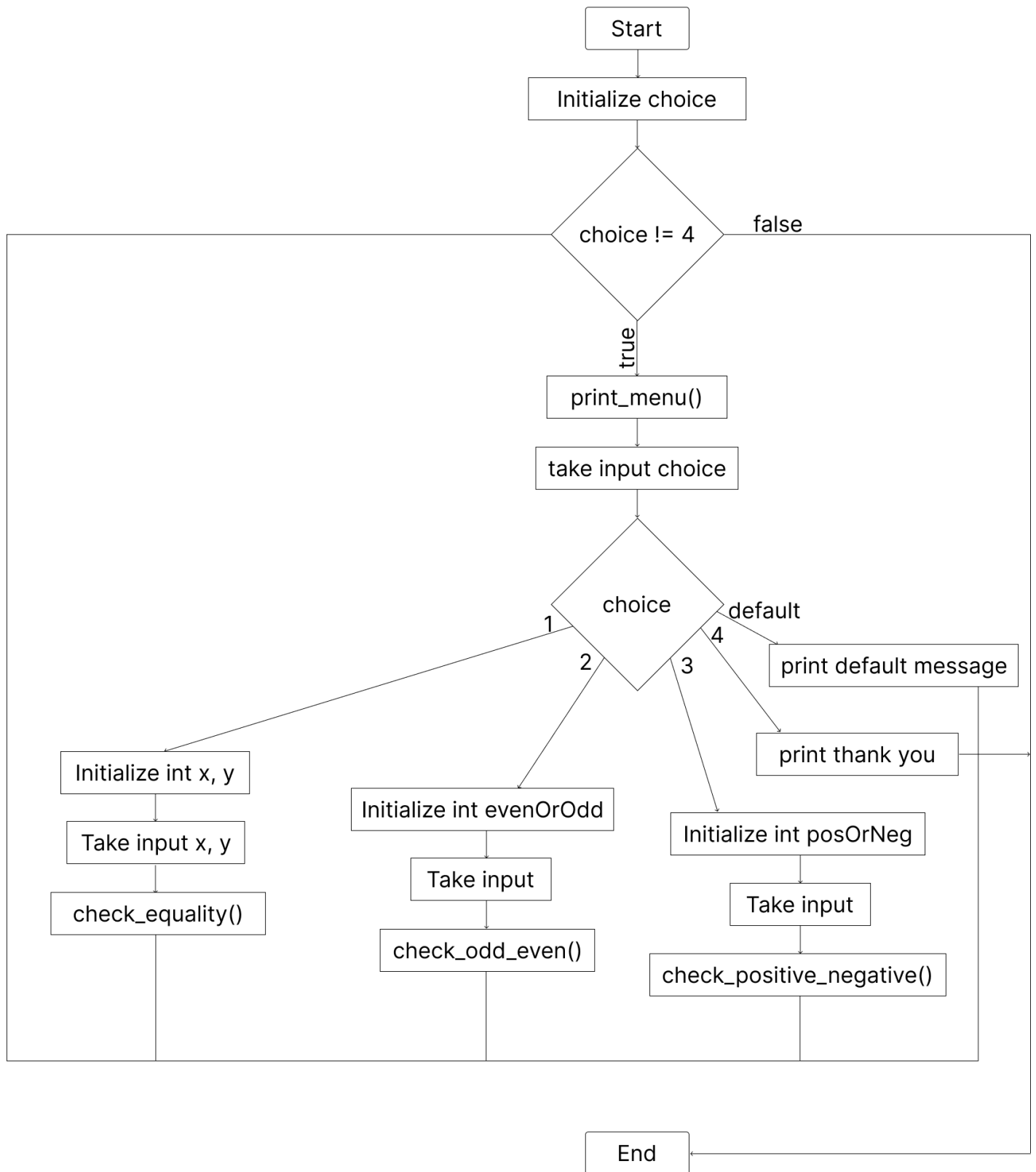
Note: Use SWITCH..CASE statement.

Ans.

Algorithm :

1. Define functions for checking equality, even/odd, and positive/negative numbers.
2. Define a function to print the main menu with available options.
3. Implement the main function :
 - a) Declare variables for choice and other necessary variables.
 - b) Enter a while loop that continues until the user selects option 4.
 - c) Inside the loop:
 - Print the menu :
 - a) Take the user's choice as input.
 - b) Use a switch statement to perform corresponding actions based on the user's choice.
 - c) For each case, take necessary inputs and call the appropriate function.
 - d) For incorrect input, display an error message.
 - d) Continue looping until the user selects option 4.
4. Exit the program.

Flowchart :



Code :

```
#include <stdio.h>
```

```
void print_menu() {  
    // Function responsible to print the initial message/menu  
    printf("Enter any number to continue \n1. Check equality of two numbers\n2.  
Check even or odd\n3. Check positive or negative numbers\n4. Quit\n\n");  
}
```

```
void check_equality(int n1, int n2) {  
    // Function responsible to check whether two numbers are equal or not  
    n1 == n2 ? printf("Both numbers are equal\n") : printf("Both numbers are not  
equal\n");  
  
    // here we're using equal to operator  
    // to check whether the variable n1 and n2 has equal value or not  
}
```

```
void check_even_odd(int n) {  
    // Function to check whether the given number is odd or even  
    n % 2 == 0 ? printf("Given number is even\n") : printf("Given number is  
odd\n");  
  
    // here we're using modulo operator to get the remainder after dividing variable  
n by 2  
    // then we're checking whether the remainder is equal to 0 or not  
    // then if the remainder is equal to 0 we're printing "Given number is even"  
    // if the remainder is not equal to 0 then we're printing "Given number is odd"  
}
```

```
void check_positive_or_negative(int n) {  
    // Function responsible to check whether the given number n is positive or  
negative
```

```
n >= 0 ? printf("Given number is positive\n") : printf("Given number is negative\n");
```

```
// here we're checking whether the value of variable n is bigger then or equal to 0 is true or not
```

```
// if it's true then we're printing "Given number is positive"
```

```
// else we're printing "Given number is negative"
```

```
}
```

```
void print_wrong_input_message() {  
    printf("Please enter correct input\n");  
}
```

```
int main() {  
    // This is The Solution of Question 1 of MCS 011 Problem Solving and  
    Programming Assignment by Sahil (2300276879)  
    int choice;
```

```
while (choice != 4) {
```

```
    // printing menu
```

```
    print_menu();
```

```
    // taking input for choice
```

```
    scanf("%d", &choice);
```

```
    // for taking decision
```

```
    switch (choice) {
```

```
        case 1: { // check equality of two numbers
```

```
            // initializing variable
```

```
            int x, y;
```

```
            // prompt for input
```

```
            printf("Enter two numbers : ");
```

```
            scanf("%d %d", &x, &y);
```

```

    // checking equality
    check_equality(x, y);
    break;
}
case 2: { // check even or odd
    // initializing variable
    int evenOrOdd;

    // taking input
    printf("Enter a number : ");
    scanf("%d", &evenOrOdd);

    // checking for odd or even
    check_even_odd(evenOrOdd);
    break;
}
case 3: { // check positive or negative
    int posOrNeg;
    printf("Enter a number : ");
    scanf("%d", &posOrNeg);
    check_positive_or_negative(posOrNeg);
    break;
}
case 4:
    printf("Thank you for using this tool");
    break;
default:
    print_wrong_input_message();
}
printf("\n\n");
}

return 0;
}

```

Q2. Write the functions in C for the following :

- (a) To find the square of any number.**
- (b) To find the absolute difference between two integers.**
- (c) To convert a decimal number to its equivalent binary number.**
- (d) To find the largest element in a given array of n elements.**

Ans.

```
#include <stdio.h>
```

```
int square(int n) {  
    return n * n;  
}
```

```
int abs_difference(int x, int y) {  
    if (x >= y) {  
        return x - y;  
    }  
  
    return y - x;  
}
```

```
long long to_binary(int decimal) {  
    if (decimal == 1 || decimal == 0) {  
        return decimal;  
    }  
  
    if (decimal == 2) {  
        return 0;  
    }  
  
    int next_binary = to_binary(decimal / 2);  
    int curr_binary = decimal % 2;  
  
    return next_binary * 10 + curr_binary;  
}
```

```

int largest_element(int *arr, int size) {
    int largest_element_index = 0;

    for (int i = 1; i < size; i++) {
        if (arr[largest_element_index] < arr[i]) {
            largest_element_index = i;
        }
    }

    return largest_element_index;
}

```

```

int main() {
    // This is The Solution of Question 2 of MCS 011 Problem Solving and
    Programming Assignment by Sahil (2300276879)

```

// Q2 (a)

```

int squareOf5 = square(5);
printf("Square of 5 is %d\n", squareOf5);
int squareOf0 = square(0);
printf("Square of 0 is %d\n", squareOf0);

```

// Q2 (b)

```

int x = 45, y = 130;
int dif = abs_difference(x, y);
printf("Difference between %d and %d is %d\n", x, y, dif);

```

// Q2 (c)

```

int binary_of_12 = to_binary(12); // 1100
int binary_of_13 = to_binary(13); // 1101
printf("binary_of_12 : %d \nbinary_of_13 : %d\n", binary_of_12, binary_of_13);

```

// Q2 (d)

```

int arr[] = {2, 5, 13, 7, 13, 1, 6};
int max_index = largest_element(arr, sizeof(arr) / sizeof(int));
printf("largest element is %d at index %d", arr[max_index], max_index);

return 0;
}

```

Q3. Write an interactive program using recursion for each of the following :

- (a) To count the digits of a given number.
- (b) To reverse a string
- (c) To find the least-common-multiple of two numbers.

Ans.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

int count_digit(int num) {
    // NOTE : This function is valid only for positive integers
    if (num < 0) {
        return -1;
    }

    if (num < 10) {
        return 1;
    }

    return 1 + count_digit(num/10);
}

```

```

void reverse_string(char *str, int start_index, int last_index) {

```



```
    if (start_index == last_index || start_index < 0 || last_index < 0 || start_index >
        strlen(str) - 1 || last_index > strlen(str) - 1 || start_index >= last_index) {
        return;
    }
```

```
    char temp = str[start_index];
    str[start_index] = str[last_index];
    str[last_index] = temp;
```

```
    reverse_string(str, start_index+1, last_index-1);
}
```

```
int lcm(int n1, int n2) {
    if (n1 == n2) {
        return n1;
    }

    if (n1 == 0 || n2 == 0) {
        return 0;
    }

    if (n1 > n2) {
        return lcm(n1-n2, n2);
    }

    return lcm(n1, n2-n1);
}
```

```
int main() {
```

```
    // This is The Solution of Question 3 of MCS 011 Problem Solving and
    Programming Assignment by Sahil (2300276879)
```

```
    // Q3 (a)
```

```
    int count = count_digit(913);
```

```
printf("Total digits : %d\n", count);
```

```
// Q3 (b)
```

```
char str[] = "ahplA";
```

```
// char str[] = "ateb";
```

```
reverse_string(str, 0, strlen(str) - 1);
```

```
printf("%s\n", str);
```

```
// Q3 (c)
```

```
int x = 4, y = 5;
```

```
int lcmOfXY = lcm(x, y);
```

```
printf("LCM of %d and %d is : %d", x, y, lcmOfXY);
```

```
return 0;
```

```
}
```

Q4. Write interactive C programs to perform the following on strings :

(a) To find the length of a given string without using the string library functions.

(b) To compare two strings without using string library functions.

(c) To count the total number of vowels and consonants in a string and display the counts separately.

Ans.

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int string_length(char *str) {
```

```
    // Function to get length of string
```

```
    int len = 0;
```

```
    while (str[len] != 0) {
```

```

        len++;
    }

    return len;
}

int compare_string(char *str, char *str2) {
    // Function to compare two strings
    int i = 0, j = 0;

    while (str[i] != 0 || str2[j] != 0) {
        if (str[i++] != str2[j++]) {
            return 0;
        }
    }

    return 1;
}

struct counter {
    // Custom data structure to store count of vowel and consonant

    int vowelCount;
    int consonantCount;
};

struct counter count_vowel_consonent(char *str) {
    // Function to compare two strings
    struct counter ctr;
    ctr.vowelCount = ctr.consonantCount = 0;
    int i = 0;

    while (str[i] != '\0') {
        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||

```

```

        str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
            ctr.vowelCount++;
        } else {
            ctr.consonantCount++;
        }

        i++;
    }

    return ctr;
}

```

```

int main() {

```

// This is The Solution of Question 4 of MCS 011 Problem Solving and Programming Assignment by Sahil (2300276879)

// Q4 (a)

```

char str[] = "Alpha";
int lenOfString = string_length(str);
printf("Length of %s is %d\n", str, lenOfString);

```

// Q4 (b)

```

// char str2[] = "Alpha";
char str2[] = "Alpha1";
if (compare_string(str, str2)) {
    printf("Both strings are equal\n");
} else {
    printf("Both strings are not equal\n");
}

```

// Q4 (c)

```

struct counter ctr = count_vowel_consonent(str);
printf("The string %s has %d vowels and %d consonants\n", str,
ctr.vowelCount, ctr.consonantCount);

```

```
    return 0;
}
```

Q5. Write an interactive C program to insert new elements in the unsorted array.
Ans.

```
#include <stdio.h>
#include <stdlib.h>
```

```
void print_message() {
    printf("Select an option:\n1. Print array\n2. Add an element at the end\n3. Add\nan element at a specific index\n0. Exit\n");
}
```

```
void print_wrong_input_message() {
    printf("Please enter a valid input\n");
}
```

```
void print_array(int *arr, int size) {
    printf("Elements currently present in the array:\n");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

```
void add_element(int **arr, int *size, int new_element) {
    // reallocating memory for new big array
    *arr = (int *)realloc(*arr, (*size + 1) * sizeof(int));
    (*arr)[*size] = new_element;
    (*size)++;
}
```

```

void insert_element(int **arr, int *size, int new_element, int index) {
    // this function will insert element at a specific position
    if (index < 0 || index > *size) {
        printf("Error: Index out of range\n");
        return;
    }

    *arr = (int *)realloc(*arr, (*size + 1) * sizeof(int));

    for (int i = *size; i > index; i--) {
        (*arr)[i] = (*arr)[i - 1];
    }

    (*arr)[index] = new_element;
    (*size)++;
}

int main() {
    // This is The Solution of Question 5 of MCS 011 Problem Solving and
    // Programming Assignment by Sahil (2300276879)
    int *arr = NULL;
    int size = 0;

    int index, elementToAdd;

    int choice = -1;

    while (choice != 0) {
        print_message();
        scanf("%d", &choice);

        switch (choice) {
            case 1:

```

```
if (size == 0) {  
    printf("There are 0 elements in the array\n\n");  
    break;  
}
```

```
print_array(arr, size);  
printf("\n");  
break;
```

case 2:

```
printf("Enter element to add : ");  
scanf("%d", &elementToAdd);  
printf("\n");
```

```
add_element(&arr, &size, elementToAdd);  
printf("\nElement added successfully\n\n");  
break;
```

case 3:

```
printf("Enter element to add : ");  
scanf("%d", &elementToAdd);  
printf("\n");  
printf("Enter index : ");  
scanf("%d", &index);
```

```
insert_element(&arr, &size, elementToAdd, index);  
printf("\nElement added successfully\n\n");  
break;
```

case 0:

```
printf("Thank you for using this tool\n");  
break;
```

default:

```
print_wrong_input_message();  
printf("\n");
```

```
}
```

```
}
```

```
free(arr);

return 0;
}
```

Q6. Using Structures in C, write an interactive program to display the mark-sheet and grade card for 10 students for a MOOC course.

Note: Assumptions can be made wherever necessary and mention them.

Ans.

```
#include <stdio.h>
```

```
#define MAX_STUDENTS 10
```

```
// structure for student grade card
```

```
struct Student {
    int _id;
    char name[50];
    int roll_number;
    int marks_subject_computer_science;
    int marks_subject_mathematics;
    int total_marks;
    char grade;
};
```

```
// Function to calculate total marks and grades
```

```
void calculate_grades(struct Student students[]) {
    for (int i = 0; i < MAX_STUDENTS; i++) {
        students[i].total_marks = students[i].marks_subject_computer_science +
students[i].marks_subject_mathematics;

        if (students[i].total_marks >= 90)
            students[i].grade = 'A';
        else if (students[i].total_marks >= 80)
```



```

        students[i].grade = 'B';
    else if (students[i].total_marks >= 70)
        students[i].grade = 'C';
    else if (students[i].total_marks >= 60)
        students[i].grade = 'D';
    else
        students[i].grade = 'F';
}
}

```

```

void print_pipes() {
    for (int i = 0; i < 45; i++) {
        printf("=");
    }
    printf("\n");
}

```

```

int main() {
    // This is The Solution of Question 6 of MCS 011 Problem Solving and
    Programming Assignment by Sahil (2300276879)
    struct Student students[MAX_STUDENTS];

```

```

    // Input student data

```

```

    printf("Enter details for %d students:\n", MAX_STUDENTS);

```

```

    for (int i = 0; i < MAX_STUDENTS; i++) {
        students[i]._id = i;

```

```

        printf("Enter for Student %d\n", i + 1);

```

```

        printf("Name : ");

```

```

        scanf("%s", students[i].name);

```

```

        printf("Roll Number : ");

```

```

        scanf("%d", &students[i].roll_number);

```

```
printf("Marks in Computer Science : ");
scanf("%d", &students[i].marks_subject_computer_science);
```

```
printf("Marks in Mathematics : ");
scanf("%d", &students[i].marks_subject_mathematics);
```

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

```
}
```

```
// Calculate total marks and grades
```

```
calculate_grades(students);
```

```
// Display the mark-sheet and grade card
```

```
print_pipes();
```

```
printf("All Marksheets and Grade Cards : \n");
```

```
for (int i = 0; i < MAX_STUDENTS; i++) {
```

```
    print_pipes();
```

```
    printf("ID : %d\n", students[i]._id);
```

```
    printf("Name : %s\n", students[i].name);
```

```
    printf("Roll Number : %d\n", students[i].roll_number);
```

```
    printf("Marks in Computer Science : %d\n",
```

```
students[i].marks_subject_computer_science);
```

```
    printf("Marks in Mathematics : %d\n",
```

```
students[i].marks_subject_mathematics);
```

```
    printf("Total Marks : %d\n", students[i].total_marks);
```

```
    printf("Grade : %c\n", students[i].grade);
```

```
    print_pipes();
```

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

```
}
```

```
return 0;
```

```
}
```

Q7. Using the File Handling concept in C programming, write the C programs for the following :

- (a) To find the number of lines in a text file.
- (b) To delete a specific line from a text file.
- (c) To copy a file to another folder with a different file-name.

Ans.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Function to print the menu
```

```
void print_menu() {
```

```
    printf("Select any option\n1. Print file\n2. Count line numbers\n3. Delete line\n4. Copy file to another file in another directory\n0. Exit\n");
}
```

```
// Function to print error message
```

```
void print_wrong_input_message() {
```

```
    printf("Please enter a valid input\n");
}
```

```
// Function to print the contents of a file
```

```
void print_file_contents(FILE *fp) {
```

```
    char ch;
    while ((ch = fgetc(fp)) != EOF) {
        printf("%c", ch);
    }
    printf("\n");
}
```

```
// Function to find the number of lines in a text file
```

```
int find_number_of_lines(FILE *fp) {
```

```
    int count = 0;
    char ch;
```

```

while ((ch = fgetc(fp)) != EOF) {
    if (ch == '\n') {
        count++;
    }
}

return count;
}

```

// Function to delete a specific line from a text file

```

void delete_line(FILE *fp, int line_number) {
    FILE *temp_fp = fopen("temp.txt", "w");
    int count = 1;
    char ch;

    while ((ch = fgetc(fp)) != EOF) {
        if (count != line_number) {
            fputc(ch, temp_fp);
        }

        if (ch == '\n') {
            count++;
        }
    }

    fclose(fp);
    fclose(temp_fp);
    remove("data.txt");
    rename("temp.txt", "data.txt");
}

```

// Function to copy a file to another folder with a different file-name

```

void copy_file(FILE *source_fp, char *destination_folder, char *new_file_name) {
    char destination_path[100];

```

```
snprintf(destination_path, sizeof(destination_path), "%s/%s",
destination_folder, new_file_name);
```

```
FILE *destination_fp = fopen(destination_path, "w");
```

```
char ch;
```

```
while ((ch = fgetc(source_fp)) != EOF) {
    fputc(ch, destination_fp);
}
```

```
fclose(source_fp);
```

```
fclose(destination_fp);
```

```
}
```

```
int main() {
```

```
// This is The Solution of Question 7 of MCS 011 Problem Solving and
Programming Assignment by Sahil (2300276879)
```

```
FILE *fp = fopen("data.txt", "r");
```

```
if (fp == NULL) {
    printf("File not found!\n");
    return 1;
}
```

```
int choice = -1;
```

```
while (choice != 0) {
    print_menu();
```

```
    scanf("%d", &choice);
```

```
    switch (choice) {
```

```
        case 1:
```

```
            print_file_contents(fp);
```

```
            fseek(fp, 0, SEEK_SET);
```

```

    break;
case 2: {
    int num_lines = find_number_of_lines(fp);
    fseek(fp, 0, SEEK_SET);
    printf("Number of Lines: %d\n", num_lines);
    break;
}
case 3: {
    int delete_line_number;
    printf("Enter the line number you want to delete: ");
    scanf("%d", &delete_line_number);
    delete_line(fp, delete_line_number);
    fp = fopen("data.txt", "r");
    break;
}
case 4: {
    char destination_folder[100];
    printf("Enter the destination folder: ");
    scanf("%s", destination_folder);

    char new_file_name[100];
    printf("Enter the new file name: ");
    scanf("%s", new_file_name);

    FILE *source_fp = fopen("data.txt", "r");
    copy_file(source_fp, destination_folder, new_file_name);
    printf("File copied successfully!\n");
    break;
}
case 0:
    printf("Exiting...\n");
    break;
default:
    print_wrong_input_message();
    break;

```

```
    }  
}  
  
fclose(fp);  
  
return 0;  
}
```

Link of all codes and resources :

<https://github.com/Sahil-4/MCS011-assignment-solution>

