Problem No: 1

Problem Name: Write a C program for a simple calculator using function.

Sample Input Output:

Case 1 ----------

input: 1

output:

Enter how many times do you want this calculator to work

1

You can use this calculator 1 time

For addition,subtraction,multiplication,division, reminder, square or cube press 1,2,3,4,5,6,7

accordingly

1

performing addition

1

1

result of addition 2.00

Opps memory is full!! Try relaunching the program

Case 2 ----------

input:: 8

output:

You can use this calculator 8 times

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

111

1

result of addition 112.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

1

performing addition

1

1

result of addition 2.00

Opps memory is full!! Try relaunching the program

Case 3 ----------

input : 2

output:

You can use this calculator 2 times

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

0

plz press between 1 to 7

For addition,subtraction,multiplication,division, reminder, percentage, square or cube press

1,2,3,4,5,6,7 accordingly

9

plz press between 1 to 7

Opps memory is full!! Try relaunching the program

Coding :

```
#include <stdio.h>

float add()

{

printf ("performing addition\n");

float a,b;

scanf ("%f%f", &a,&b);

return a+b;

}

float sub()

{

printf ("performing subtraction\n");

float a,b;

scanf ("%f%f",&a, &b);
```

```
return a-b;
}
float mul()
{
printf ("performing multiplication\n");
float a,b;
scanf ("%f%f", &a,&b);
return a*b;
}
float div()
{
printf ("performing division\n");
float a,b;
scanf ("%f%f",&a,&b);
if (b==0)
{
printf ("can not be divided by 0 and result will be wrong\n");
}
else
{
return a/b;
}
}
int rem()
{
printf ("performing reminder\n");
int a,b;
scanf ("%d%d",&a,&b);
return a%b;
}
float squ()
{
printf ("performing square\n");
float a;
scanf ("%f",&a);
```

```
float b;
b=a*a;
return b;
}
float cub()
{
printf ("performing cube\n");
float a;
scanf ("%f", &a);
float b;
b=(a*a)*a;
return b;
}
int main ()
{
printf ("Enter how many times do you want this calculator to work\n");
int x;
scanf ("%d",&x);
if (x==1)
{printf ("You can use this calculator %d time\n", x);}
else
{
printf ("You can use this calculator %d times\n",x);
}
for ( int y=1;y<=x;y++)
{
printf ("For addition,subtraction,multiplication,division, reminder, square or cube press
1,2,3,4,5,6 or 7 accordingly \n");
int a;
scanf ("%d",&a);
if (a>8)
{
printf ("plz press between 1 to 8\n");
}
else if (a<=0)
```

```
{
printf ("plz press between 1 to 8\n");
}
else if (a==1)
{
printf ("result of addition %.2f\n",add());
}
else if (a==2)
{
printf ("result of subtraction %.2f\n", sub());
}
else if (a==3)
{
printf ("result of multiplication %.2f\n", mul());
}
else if (a==4)
{
printf ( "result of division %.2f\n", div());
}
else if (a==5)
{
printf ("result of reminder %d\n", rem());
}
else if (a==6)
{
printf ("result of square %.2f\n", squ());
}
else
{
printf ("result of cube %.2f\n",cub());
}
}
printf ("\nOpps memory is full!! Try relaunching the program\n");
return 0;}
```

Result and Analysis :

The program enables user to perform how many times he\she can use the calculator. User will be asked to enter between 1 to 8 values to perform addition, subtraction, multiplication, division, reminder, percentage, square or cube respectively.

If the user enters the required value then the code will execute its equivalent function and print the value. Noted that if the dividend value entered is 0 the the code will show error and the result will be error.

After performing the task, the program will show “memory is full” message and will be executed from the program.

Problem NO: 2

Problem Name: Solve a problem of any OJ (Online Judge) using both pointer and normal way.

Adam and Chef have some number of sticks and they want to swap their sticks.

Sample Input Output:

Input Output

1 2 Values after swapping using pointers:

num1: 2

num2: 1

Values after swapping without using pointers:

num1: 2

num2: 1

source code:

```
#include <stdio.h>

void swapWithPointers(int *a, int *b)

{

int temp = *a;

*a = *b;

*b = temp;

}

void swapWithoutPointers(int a, int b)

{

int temp = a;

a = b;

b = temp;

}

int main() {

int num1, num2;
```

```
scanf("%d%d", &num1,&num2);
swapWithPointers(&num1, &num2);
printf("\nValues after swapping using pointers:\n");
printf("num1: %d\n", num1);
printf("num2: %d\n", num2);
swapWithoutPointers(num1, num2);
printf("\nValues after swapping without using pointers:\n");
printf("num1: %d\n", num1);
printf("num2: %d\n", num2);
return 0;
}
```

Result and Analysis:

When swapping with pointers the values of the variables return the addresses of the value and when swapping without pointers or normal way the function directly returns the value.

Problem NO: 3

problem Name: Write a C program to check whether a number is palindrome or not.

Sample Input Output:

Input	Output
123	Not Palindrome
121	palindrome

Source Code:

```
#include<stdio.h>
int palindrome( int a)
{
int reminder, reversednumber=0;
int originalnumber=a;
while (a>0)
{
reminder=a%10;
reversednumber= reversednumber*10+ reminder;
a= a/10;
}
if (originalnumber==reversednumber)
{
return 1;
```

```
}
else
{
return 0;
}
}
int main()
{
int a;
scanf ("%d",&a);
if (palindrome(a))
{
printf ("Palindrome");
}
else
{
printf ("Not Palindrome");
}
return 0;
}
```

Result and Analysis:

User will input a number and the code will check whether it is palindrome or not.

The palindrome function will check the original number and whether the reverse number is the same or not. If it is the same, then it will return true otherwise false. In the main function if the sub function is true, it will execute properly

Problem NO:4

Problem Name: Write a program to find the sum of all digits of a large number.

Input Output

123 6

Source Code:

```
#include <stdio.h>
#include <stdlib.h>
int main() {
char number[1000];
```

```
int sum = 0;
fgets(number, sizeof(number), stdin);
for (int i = 0; number[i] != '\0'; i++) {
if (isdigit(number[i]))
{
sum += number[i] - '0';
}
}
printf("%d\n", sum);
return 0;
}
```

Result and Analysis:

This program takes input as character string and checks it it's a digit using isdigit function and finaly converts it into integer and sum the digits.

Problem NO: 5

Problem Name: Write a c program to check if the number is palindrome or not using recursion.

Input	Output
121	Palindrome
123	Not Palindrome

Source Code:

```
#include <stdio.h>
int isPalindrome(int num, int originalNum, int reversedNum) {
if (num == 0) {
if (originalNum == reversedNum) {
return 1;
} else {
return 0;
}
} else {
int lastDigit = num % 10;
reversedNum = reversedNum * 10 + lastDigit;
return isPalindrome(num / 10, originalNum, reversedNum);
}
}
```

```
int main() {
int num1;
scanf("%d", &num1);
if (isPalindrome(num1,num1,0)) {
printf("Palindrome");
}
else {
printf("Not Palindrome");
}
return 0;
}
```

Result and Analysis:

In this code the ispalindrome function takes three parameters and calls the function recursively

itself in return and reverse the digits if it matches the original number.

Problem No: 6

Problem Name: Write a program to find the sum of all digits using recursion.

Input Output

123 6

12 3

Source Code:

```
#include <stdio.h>
int sumOfDigits(int num) {
if (num == 0)
return 0;
else
return num % 10 + sumOfDigits(num / 10);
}
int main() {
int num;
scanf("%d", &num);
int result = sumOfDigits(num);
printf("%d\n",result);
return 0;
}
```

Result and Analysis:

This program takes one parameter in subfunction and the subfunction recursively returns the function to get sum of the digits and finally when the subfunction is being called in main function the result is being shown.

Problem No: 7

Problem Name: Write a C program to generate nth Fibonacci term using recursion.

Input Output

7 13

0 0

Source Code:

```
#include <stdio.h>
int fibonacci(int n) {
if (n <= 1) {
return n;
} else {
return fibonacci(n - 1) + fibonacci(n - 2);
}
}
int main() {
int n;
scanf("%d", &n);
printf("%d\n",fibonacci(n));
return 0;
}
```

Result and Analysis:

The subfunction takes one value as integer and in return it calls the function itself. Finally in main function the subfunction is called and give us the nth value of Fibonacci.

Problem No: 8

Problem Name: Write a c program to find the GCD of two numbers using recursion.

Input Output

2 2

4

70

100 10

Source Code:

```
#include <stdio.h>
int gcd(int a, int b) {
if (b == 0) {
return a;
} else {
return gcd(b, a % b);
}
}
int main() {
int num1, num2;
scanf("%d%d", &num1,&num2);
printf("%d\n",gcd(num1, num2));
return 0;
}
```

Result and Analysis:

This program takes two integers in subfunction called gcd and calls itself recursively in return.

Finally when it is called in main function then it shows the result.

Problem No: 9

Problem Name: Write a c program to find the LCM of two numbers using Recursion.

Input Output

70

100 700

Source Code:

```
#include <stdio.h>
int gcd(int a, int b) {
if (b == 0) {
return a;
} else {
return gcd(b, a % b);
}
}
int lcm(int a, int b) {
```

```
return (a * b) / gcd(a, b);
}
int main() {
int num1, num2;
scanf("%d%d", &num1,&num2);
printf("%d\n", lcm(num1, num2));
return 0;
}
```

Result and Analysis:

In the gcd subfunction it returns itself recursively and, in the lcm subfunction it received two integers as well as the gcd subfunction it recursively calls the gcd subfunction again to show the result in mainfunction

Problem No: 10

Problem Name: Write a c program to find the sum of an array using recursion.

Input Output

3

2 3 4 9

Source Code:

```
#include <stdio.h>
int arraySum(int arr[], int size) {
if (size == 0) {
return 0;
}
return arr[size - 1] + arraySum(arr, size - 1);
}
int main() {
int size;
scanf("%d", &size);
int arr[size];
for (int i = 0; i < size; i++) {
scanf("%d", &arr[i]);
}
printf("%d\n", arraySum(arr, size));
return 0;
```

}

Result and Analysis:

In subfunction the array is returned to the subfunction itself to sum the array elements.

In the main function user must input the size of the array and enter the elements using space. The subfunction will be called in the main function and show the sum of the elements.

Problem No: 11

Problem Name: Write a c program to reverse an array using pointer.

Input Output

4 4 3 2 1

1 2 3 4

Source Code:

```
#include <stdio.h>
void reverseArray(int *arr, int size) {
int *start = arr;
int *end = arr + size - 1;
while (start < end) {
int temp = *start;
*start = *end;
*end = temp;
start++;
end--;
}
}
int main() {
int size;
scanf("%d", &size);
int arr[size];
for (int i = 0; i < size; i++) {
scanf("%d", &arr[i]);
}
reverseArray(arr, size);
for (int i = 0; i < size; i++) {
printf("%d ", arr[i]);
}
```

```
return 0;
}
```

Result and Analysis:

This program uses a void function to reverse the array. When the subfunction is called in the main function then the output is shown.

Problem No:

Problem Name: Write a c program to copy an array using pointer.

Input Output

4 Copied Array

1 2 3 4 1 2 3 4

Source code:

```
#include <stdio.h>
void copyArray(int *source, int *destination, int size) {
for (int i = 0; i < size; i++) {
*(destination + i) = *(source + i);
}
}
int main() {
int size;
scanf("%d", &size);
int sourceArray[size];
int destinationArray[size];
for (int i = 0; i < size; i++) {
scanf("%d", &sourceArray[i]);
}
copyArray(sourceArray, destinationArray, size);
printf("Copied array:\n");
for (int i = 0; i < size; i++) {
printf("%d ", destinationArray[i]);
}
return 0;
}
```

Result and Anaysis:

This program uses a void function to copy an array using pointer. When the subfunction is

called in the main function then the result is shown.

Problem No: 13

Problem Name: Create structure that stores a student's name, id, cgpa and sort the structure using bubble sort algorithm and output the final structure.

Input

Enter the number of students: 2

Enter details for student 1:

Name: rahat

ID: 1045

CGPA: 3.58

Enter details for student 2:

Name: ratul

ID: 1055

CGPA: 3.44

Output

Sorted student details based on CGPA:

Name ID CGPA

rahat 1045 3.58

ratul 1055 3.44

Source Code

```
#include <stdio.h>
#include <string.h>
struct Student {
char name[50];
int id;
float cgpa;
};
void bubbleSort(struct Student arr[], int n) {
int i, j;
struct Student temp;
for (i = 0; i < n - 1; i++) {
for (j = 0; j < n - i - 1; j++) {
if (arr[j].cgpa < arr[j + 1].cgpa) {
```

```
temp = arr[j];
arr[j] = arr[j + 1];
arr[j + 1] = temp;
}
}
}
}
int main() {
int n;
printf("Enter the number of students: ");
scanf("%d", &n);
struct Student students[n];
for (int i = 0; i < n; i++) {
printf("Enter details for student %d:\n", i + 1);
printf("Name: ");
scanf("%s", students[i].name);
printf("ID: ");
scanf("%d", &students[i].id);
printf("CGPA: ");
scanf("%f", &students[i].cgpa);
}
bubbleSort(students, n);
printf("\nSorted student details based on CGPA:\n");
printf("%-10s %-10s %-10s\n", "Name", "ID", "CGPA");
for (int i = 0; i < n; i++) {
printf("%-10s %-10d %-10.2f\n", students[i].name, students[i].id, students[i].cgpa);
}
return 0;
}
```

Result and Analysis:

This program first constructs a structure of students. Then bubble sort the details in the end. Meanwhile in main function the structure is made sorted by calling the subfunction and finally shows output in sorted way with respect to CGPA .

Problem NO: 14

Problem Name: Write a c program to implement grading system using macros.

Input output

50 F

60 D

95 A

Source Code:

```
#include <stdio.h>

#define A_GRADE 90

#define B_GRADE 80

#define C_GRADE 70

#define D_GRADE 60

#define GET_GRADE(score) \

(score >= A_GRADE) ? 'A' : \

(score >= B_GRADE) ? 'B' : \

(score >= C_GRADE) ? 'C' : \

(score >= D_GRADE) ? 'D' : 'F'

int main() {

int studentScore;

scanf("%d", &studentScore);

printf("%c\n", GET_GRADE(studentScore));

return 0;

}
```

Result and Analysis:

In this program there are four macros defined. They check the condition if it's true or not then in the main function the function of defined macro is being called and shows the output of included grade.

Problem No: 15

Problem Name: Write a C program to store information of student by handing files and modifying data.

Input

Menu:

1. Add student

2. Display all students

3. Modify student data

4. Exit

Enter your choice:

Ouput

Enter your choice: 1

Enter student name: rahat

Enter roll number: 1045

Enter marks: 60

Student record added successfully.

Source Code:

```
#include <stdio.h>

#include <stdlib.h>

struct Student {

char name[50];

int rollNumber;

float marks;

};

void addStudent(FILE *file) {

struct Student newStudent;

printf("Enter student name: ");

scanf("%s", newStudent.name);

printf("Enter roll number: ");

scanf("%d", &newStudent.rollNumber);

printf("Enter marks: ");

scanf("%f", &newStudent.marks);

fwrite(&newStudent, sizeof(struct Student), 1, file);

printf("Student record added successfully.\n");

}

void displayStudents(FILE *file) {

struct Student currentStudent;

rewind(file);

while (fread(&currentStudent, sizeof(struct Student), 1, file) == 1) {

printf("Name: %s\n", currentStudent.name);

printf("Roll Number: %d\n", currentStudent.rollNumber);
```

```
printf("Marks: %.2f\n", currentStudent.marks);
printf("\n");
}
}
void modifyStudent(FILE *file, int rollNumber) {
struct Student currentStudent;
long int position;
rewind(file);
while (fread(&currentStudent, sizeof(struct Student), 1, file) == 1) {
if (currentStudent.rollNumber == rollNumber) {
position = ftell(file) - sizeof(struct Student);
break;
}
}
if (feof(file)) {
printf("Student with roll number %d not found.\n", rollNumber);
return;
}
printf("Enter new name: ");
scanf("%s", currentStudent.name);
printf("Enter new marks: ");
scanf("%f", &currentStudent.marks);
fseek(file, position, SEEK_SET);
fwrite(&currentStudent, sizeof(struct Student), 1, file);
printf("Student record modified successfully.\n");
}
int main() {
FILE *file;
int choice, rollNumber;
file = fopen("student_records.dat", "rb+");
if (file == NULL) {
file = fopen("student_records.dat", "wb+");
}
do {
printf("\nMenu:\n");
```

```
printf("1. Add student\n");
printf("2. Display all students\n");
printf("3. Modify student data\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
case 1:
addStudent(file);
break;
case 2:
displayStudents(file);
break;
case 3:
printf("Enter the roll number of the student to modify: ");
scanf("%d", &rollNumber);
modifyStudent(file, rollNumber);
break;
case 4:
printf("Exiting program.\n");
break;
default:
printf("Invalid choice. Please enter a valid option.\n");
}
} while (choice != 4);
fclose(file);
return 0;
}
```

Result and Analysis:

This program uses structure to add student information. It uses file to point for arguments. The rewind function is used for indicating the pointer to the beginning of a file. It uses fread to re-read the file and fseek to re seek the file and creates a menu folder to make choice for the user.