Week 10

Q1)

main.c

Share Run

```c
#include <stdio.h>

int sumOfDigits(int num) {
    if (num == 0) {
        return 0;
    }
    return (num % 10) + sumOfDigits(num / 10);
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);

    int result = sumOfDigits(number);
    printf("Sum of digits: %d\n", result);

    return 0;
}
```

Output

```
Enter a number: 2 4 3
Sum of digits: 2


=== Code Execution Successful ===
```

Q2)

main.c

Share Run

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

void reverseString(char str[], int start, int end) {
    if (start >= end) {
        return;
    }
    char temp = str[start];
    str[start] = str[end];
    str[end] = temp;
    reverseString(str, start + 1, end - 1);
}

int main() {
    char str[100];
    printf("Enter a string: ");
    scanf("%s", str);

    int len = strlen(str);
    reverseString(str, 0, len - 1);

    printf("Reversed string: %s\n", str);

    return 0;
}
```

Output

```
Enter a string: attacool
Reversed string: loocatta


=== Code Execution Successful ===
```

Q3)

main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
typedef struct flight
{
    int flight_number;
    char dest[30];
    char dep[30];
    char date[30];
    int seats;
}flight;
int book_a_seat(flight* f)
{
    if(f->seats==0)
    {
    printf("All seats are booked");
    return 0;
    }
    else
    {
    printf("Enter flight number: ");
    scanf("%d",&f->flight_number);
    printf("Enter destination city: ");
    scanf(" %[^\n]",f->dest);
    printf("Enter departure city: ");
    scanf(" %[^\n]",f->dep);
    printf("Enter date : ");
    scanf(" %[^\n]",f->date);
    printf("Flight booked succesfully!\n\n");
    f->seats-=1;
    return 1;
    }
}
```

Output

```
How many seats are available?: 5
Enter flight number: 22
Enter destination city: karachi
Enter departure city: lahore
Enter date : 22/12/24
Flight booked succesfully!

Flight Number: 22
Destination City: karachi
Departure City: lahore
Date:22/12/24
Seats available: 4

=== Code Execution Successful ===
```

```c
void display(flight f)
{
    printf("Flight Number: %d\n",f.flight_number);
    printf("Destination City: %s\n",f.dest);
    printf("Departure City: %s\n",f.dep);
    printf("Date:%s \n",f.date);
    printf("Seats available: %d",f.seats);
}
int main()
{
    flight f1;
    printf("How many seats are available?: ");
    scanf("%d",&f1.seats);
    if(book_a_seat(&f1)!=0)
    {
        display(f1);
    }
```

Q4)

main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
#define mcount 100
typedef struct movie
{
    char title[50];
    char genre[20];
    char director[30];
    int release_year;
    float rating;
}movie;
void add_movies(movie* m,int* ind)
{
    if(*ind>=mcount)
    {
        printf("maximum movies added!");
        return;
    }
    else
    {
    printf("Enter the title of the movie: ");
    scanf(" %[^\n]",m[*ind].title);
    printf("Enter the genre of the movie: ");
    scanf(" %[^\n]",m[*ind].genre);
    printf("Director of the movie: ");
    scanf(" %[^\n]",m[*ind].director);
    printf("Enter the release year of the movie: ");
    scanf("%d",&m[*ind].release_year);
    printf("Enter the ratings of the movie: ");
    scanf("%f",&m[*ind].rating);
    printf("movie added succesfully! ");
    (*ind)++;
    }
}
void display(movie* m,int ind)
{
    int i;
    for(i=0;i<ind;i++)
    {
        printf("\nRecord %d",i+1);
        printf("\nTitle: %s\n",m[i].title);
        printf("Genre: %s\n",m[i].genre);
        printf("Director: %s\n",m[i].director);
        printf("Release Year: %d\n",m[i].release_year);
        printf("Ratings: %f\n",m[i].rating);
    }
}
void updatebygenre(movie* m,int ind)
{
    int i,found;
    char g[20];
    printf("Search by genre: ");
    scanf(" %[^\n]",g);
    for(i=0;i<ind;i++)
    {
        found=0;
        if(strcmp(m[i].genre,g)==0)
        {
            found=1;
        }
    if(found)
    {
        printf("\nRecord %d",i+1);
        printf("\nTitle: %s\n",m[i].title);
        printf("Genre: %s\n",m[i].genre);
        printf("Director: %s\n",m[i].director);
        printf("Release Year: %d\n",m[i].release_year);
        printf("Ratings: %f\n",m[i].rating);
    }
```

Output

```
Choose from following:
1. Add Record
2. Search by Genre
3. Display all records
Any other key to exit
choice: 1
Enter the title of the movie: Conjuring
Enter the genre of the movie: Horror
Director of the movie: Naheel
Enter the release year of the movie: 2016
Enter the ratings of the movie: 8.8
movie added succesfully!

Choose from following:
1. Add Record
2. Search by Genre
3. Display all records
Any other key to exit
choice: 2
Search by genre: Horror

Record 1
Title: Conjuring
Genre: Horror
Director: Naheel
Release Year: 2016
Ratings: 8.800000


Choose from following:
1. Add Record
2. Search by Genre
3. Display all records
Any other key to exit
choice: 3
```

```
    }
    if(!found)
        printf("Record not found!\n");
}
int main()
{
    int ind=0;
    movie m[mcount];
    int ch;
    while(1)
    {
    printf("Choose from following: \n");
    printf("1. Add Record\n");
    printf("2. Search by Genre\n");
    printf("3. Display all records\n");
    printf("Any other key to exit\n");
    printf("choice: ");
    scanf("%d",&ch);
    if(ch<1 || ch>3)
    {
        break;
    }
    switch(ch)
    {
        case 1:
        add_movies(m,&ind);
        printf("\n\n");
        break;
        case 2:
        updatebygenre(m,ind);
        printf("\n\n");
        break;
        case 3:
        display(m,ind);
        printf("\n\n");
```

Q5)

main.c

Share Run

```c
#include <stdio.h>

void printArrayRecursive(int arr[], int size) {
    if (size == 0) {
        return;
    }
    printf("%d ", arr[0]);
    printArrayRecursive(arr + 1, size - 1);
}

int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];
    printf("Enter %d elements: ", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Array elements are: ");
    printArrayRecursive(arr, size);

    return 0;
}
```

Output

```
Enter the size of the array: 3
Enter 3 elements: 2 4 5 3 6
Array elements are: 2 4 5

=== Code Execution Successful ===
```

Q6)

main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
#include<math.h>
typedef struct points
{
    float x;
    float y;
}points;
float distance(points p1,points p2)
{
    float dist=sqrt(pow(p1.x-p2.x,2)+pow(p1.y-p2.y,2));
    return dist;
}
void check(points test,points bottom,points top)
{
    float xmax,xmin;
    float ymax,ymin;
    if(bottom.x>=top.x)
    {
        xmax=bottom.x;
        xmin=top.x;
    }
    else if(bottom.x<=top.x)
    {
        xmin=bottom.x;
        xmax=top.x;
    }

    if(bottom.y>=top.y)
    {
        ymax=bottom.y;
        ymin=top.y;
    }
    else if(bottom.y<=top.y)
    {
```

Output

```
Enter coordinates of x and y for point 1: 2 4
Enter coordinates of x and y for point 2: 3 7
Distance: 3.162278
Enter test point(x,y): 5 3
Enter bottom left (x,y): 2 6
Enter top right (x,y): 1 4
point (5.00,3.00) does not lies in the rectangular boundary


=== Code Execution Successful ===
```

```c
        ymin=bottom.y;
        ymax=top.y;
    }

    if((test.x>=xmin && test.x<=xmax)&&(test.y>=ymin && test.y<=ymax))
    {
        printf("point (%.2f,%.2f) lies in the rectangular boundary\n",test.x,test.y);
    }
    else
    {
        printf("point (%.2f,%.2f) does not lies in the rectangular boundary\n",test.x,test
            .y);
    }

}
int main()
{
    points p1,p2;
    points test,bottom,top;
    printf("Enter coordinates of x and y for point 1: ");
    scanf("%f %f",&p1.x,&p1.y);
    printf("Enter coordinates of x and y for point 2: ");
    scanf("%f %f",&p2.x,&p2.y);
    printf("Distance: %f\n",distance(p1,p2));
    printf("Enter test point(x,y): ");
    scanf("%f %f",&test.x,&test.y);
    printf("Enter bottom left (x,y): ");
    scanf("%f %f",&bottom.x,&bottom.y);
    printf("Enter top right (x,y): ");
    scanf("%f %f",&top.x,&top.y);
    check(test,bottom,top);
```

Q7)

main.c

Share Run

```c
#include<stdio.h>
#define MAX_TEMP 100
int check()
{
    static int count=0;
    count++;
    return count;
}
int main()
{

    int n,i,store=0;
    printf("How many temperatures do you want to enter?: ");
    scanf("%d",&n);
    float arr[n];
    printf("Enter temps:\n");
    for(i=0;i<n;i++)
    {
        scanf("%f",&arr[i]);
        if(arr[i]>MAX_TEMP)
        {
            store=check();
        }
    }
    printf("\n%d times temperature exceeded the max temperature",store);
    return 0;
}
```

Output

```
How many temperatures do you want to enter?: 3
Enter temps:
98 100 102

1 times temperature exceeded the max temperature

=== Code Execution Successful ===
```

Q8)

main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
#define maxind 100
typedef struct cars
{
    char make[30];
    char model[30];
    int year;
    float price;
    float mileage;
}cars;
void add(cars* c1,int* i)
{
    if(*i>maxind)
    {
        printf("Maximum cars added!");
    }
    else
    {
    printf("Enter make of the car: ");
    scanf(" %[^\n]",c1->make);
    printf("Enter model of the car: ");
    scanf(" %[^\n]",c1->model);
    printf("Enter release year: ");
    scanf("%d",&c1->year);
    printf("Enter price in usd: ");
    scanf("%f",&c1->price);
    printf("Enter mileage: ");
    scanf("%f",&c1->mileage);
    (*i)++;
    }
}
```

Output

```
Enter your choice from following:
1. Add a record
2. Display Record
3. Search by make or model
Any other button to exit
Choice: 1
Enter make of the car: German
Enter model of the car: X
Enter release year: 2019
Enter price in usd: 2500
Enter mileage: 450

Enter your choice from following:
1. Add a record
2. Display Record
3. Search by make or model
Any other button to exit
Choice: 3

Search by
1.Make
2.Model
Choice: 2

Enter model: X
Car 1 data
make of the car: German
model of the car:X
release year:2019
price in usd:2500.000
mileage:450.000
```

```c
void display(cars* c1,int ind)
{
    int i;
    for(i=0;i<ind;i++)
    {
    printf("\nCar %d data\n",i+1);
    printf("make of the car: %s\n",c1[i].make);
    printf("model of the car:%s\n",c1[i].model);
    printf("release year:%d\n",c1[i].year);
    printf("price in usd:%.3f\n",c1[i].price);
    printf("mileage:%.3f\n",c1[i].mileage);
    }
}
void search(cars* c1,int ind,int ch)
{
    int found=0,i;
    if(ch==1)
    {
        char mk[30];
        printf("\nEnter make: ");
        scanf(" %[^\n]",mk);
        for(i=0;i<ind;i++)
        {
            if(strcmp(c1[i].make,mk)==0)
            {
                found=1;
            }
            if(found)
            {
                printf("Car %d data\n",i+1);
                printf("make of the car: %s\n",c1[i].make);
                printf("model of the car:%s\n",c1[i].model);
                printf("release year:%d\n",c1[i].year);
                printf("price in usd:%.3f\n",c1[i].price);
```

```
            printf("mileage:%.3f\n",c1[i].mileage);
        }
    }
    if(!found)
    {
        printf("Make not found!\n");
    }
}
else if(ch==2)
{
    found=0;
    char mod[30];
    printf("\nEnter model: ");
    scanf(" %[^\n]",mod);
    for(i=0;i<ind;i++)
    {
        if(strcmp(c1[i].model,mod)==0)
        {
            found=1;
        }
        if(found)
        {
            printf("Car %d data\n",i+1);
            printf("make of the car: %s\n",c1[i].make);
            printf("model of the car:%s\n",c1[i].model);
            printf("release year:%d\n",c1[i].year);
            printf("price in usd:%.3f\n",c1[i].price);
            printf("mileage:%.3f \n",c1[i].mileage);
        }
    }
    if(!found)
    {
        printf("Model not found!\n");
    }
```

```
    }
    else
    printf("Invalid choice!\n");
}
int main()
{
    cars c[maxind];
    int i=0;
    while(1)
    {
    printf("\nEnter your choice from following:\n");
    printf("1. Add a record\n");
    printf("2. Display Record\n");
    printf("3. Search by make or model\n");
    printf("Any other button to exit");
    printf("\nChoice: ");
    int choice;
    scanf("%d",&choice);
    if(choice<1 || choice>3)
    return 0;
    switch(choice)
    {
        case 1:
        add(c,&i);
        break;
        case 2:
        display(c,i);
        break;
        case 3:
        printf("\nSearch by\n1.Make\n2.Model\nChoice: ");
        scanf("%d",&choice);
        search(c,i,choice);
    }
    }
    return 0;
```

Q9)

main.c

Share Run

```c
#include <stdio.h>

void bubbleSortRecursive(int arr[], int n) {
    if (n == 1) {
        return;
    }

    for (int i = 0; i < n - 1; i++) {
        if (arr[i] > arr[i + 1]) {
            int temp = arr[i];
            arr[i] = arr[i + 1];
            arr[i + 1] = temp;
        }
    }

    bubbleSortRecursive(arr, n - 1);
}

int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];
    printf("Enter %d elements: ", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    bubbleSortRecursive(arr, size);

    printf("Sorted array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
```

Output

```
Enter the size of the array: 4
Enter 4 elements: 24 47 8 15
Sorted array: 8 15 24 47

=== Code Execution Successful ===
```

Q10)

main.c

Share Run

```c
#include<stdio.h>
#include<string.h>
#define maxind 100
typedef struct travel
{
    char name[30];
    char dest[30];
    float duration;
    float cost;
    int seats;
}travel;
void add(travel* t,int* i)
{
    if(*i>maxind)
    {
        printf("\nMaximum cars added!");
    }
    else
    {
    printf("\nEnter name of the package: ");
    scanf(" %[^\n]",t->name);
    printf("Enter destination: ");
    scanf(" %[^\n]",t->dest);
    printf("Enter duration: ");
    scanf("%f",&t->duration);
    printf("Enter cost in usd: ");
    scanf("%f",&t->cost);
    printf("Enter seats available: ");
    scanf("%d",&t->seats);
    (*i)++;
    }
}
void display(travel* t,int i)
{
    int ind;
```

Output

```
Choose from the following:
1. Add records
2. Display all Records
3. Book a package
Other key to exit
Choice: 1

Enter name of the package: Premium
Enter destination: Las Vegas
Enter duration: 3
Enter cost in usd: 110
Enter seats available: 6

Choose from the following:
1. Add records
2. Display all Records
3. Book a package
Other key to exit
Choice: 3

Enter the name of package to book: Premium
Package "Premium" booked

Choose from the following:
1. Add records
2. Display all Records
3. Book a package
Other key to exit
Choice: 2

Package 1
Name of the package: Premium
Destination: Las Vegas
Duration: 3.000000
Cost in usd: 110.000000
```

```c
    for(ind=0;ind<i;ind++)
    {
    printf("\nPackage %d\n",ind+1);
    printf("Name of the package: %s\n",t[ind].name);
    printf("Destination: %s\n",t[ind].dest);
    printf("Duration: %f\n",t[ind].duration);
    printf("Cost in usd: %f\n",t[ind].cost);
    printf("Seats available: %d\n",t[ind].seats);
    }
}
void book(travel* t,int ind)
{
    char name[30];
    int i,found;
    printf("\nEnter the name of package to book: ");
    scanf(" %[^\n]",name);
    for(i=0;i<ind;i++)
    {
        if(strcmp(t->name,name)==0)
        {
            found=1;
            break;
        }
    }
    if(found)
    {
    if(t[i].seats==0)
    {
        printf("Max seats booked!\n");
    }
    else
    {
```

```
68          t->seats-=1;
69          printf("Package \"%s\" booked\n",t[i].name);
70      }
71      }
72  }
73  int main()
74  {
75      travel t1[maxind];
76      int index=0;
77      int choice;
78      while(1)
79      {
80          printf("\nChoose from the following:\n");
81          printf("1. Add records\n");
82          printf("2. Display all Records\n");
83          printf("3. Book a package\n");
84          printf("Other key to exit\n");
85          printf("Choice: ");
86          scanf("%d",&choice);
87          switch(choice)
88          {
89              case 1:
90              add(t1,&index);
91              break;
92              case 2:
93              display(t1,index);
94              break;
95              case 3:
96              book(t1,index);
97              break;
98              default:
99              return 0;
100         }
101     }
102     return 0;
```

Q11)

main.c

```c
#define conv 0.001
int mkm(int meter)
{
    static int count=0;
    if(meter!=0)
    {
    printf("Kilometers: %.3f\n",meter*conv);
    count++;
    }
    return count;
}
int main()
{
    int ch;
    int m,km;
    while (1)
    {
        printf("Choose from following: \n");
        printf("1. Meters to kilometers\n");
        printf("Any other key to exit\n");
        printf("Choice: ");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
            printf("Enter distance in meters: ");
            scanf("%d",&m);
            mkm(m);
            break;
            default:
            printf("Function was called %d times",mkm(0));
            return 0;
        }
    }
    printf("Function was called %d times",mkm(0));
```

Output

```
Choose from following:
1. Meters to kilometers
Any other key to exit
Choice: 1
Enter distance in meters: 1500
Kilometers: 1.500
Choose from following:
1. Meters to kilometers
Any other key to exit
Choice:
```

Q12)

main.c

```c
#include <stdio.h>

int linearSearch(int arr[], int size, int target, int index) {
    if (index >= size) {
        return -1; // Target not found
    }
    if (arr[index] == target) {
        return index; // Target found
    }
    return linearSearch(arr, size, target, index + 1); // Continue searching
}

int main() {
    int size, target;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];
    printf("Enter %d elements: ", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter the target element to search for: ");
    scanf("%d", &target);

    int result = linearSearch(arr, size, target, 0);
    if (result != -1) {
        printf("Element found at index %d\n", result);
    } else {
        printf("Element not found in the array\n");
    }

    return 0;
}
```

Output

```
Enter the size of the array: 4
Enter 4 elements: 5 12 26 8
Enter the target element to search for: 26
Element found at index 2


=== Code Execution Successful ===
```