Assignment 14 || Structure

Arjun Patel – FRN-13J1124/006

Note: I mistakenly not uploaded part 2 and 3 of assignment 13 so, I am submitting it here.

Assignment 13 remaining part:

Q1) Pass by Value using function

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

#define s(x) x*x

typedef struct Employee{
    int id;
    char name[30];
    int salary;
} Employee;

void storeEmps(Employee* emp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter id\n");
        scanf("%d", &emp[i].id);
        printf("Enter name\n");
        scanf("%s", emp[i].name);
        printf("Enter salary\n");
        scanf("%d", &emp[i].salary);
    }
}
void printEmps(Employee* emp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\nid-> %d  Name-> %s  Salary->%d\n", emp[i].id, emp[i].name,
emp[i].salary);
    }
}


int main(){
    Employee arr[3];
    storeEmps(arr, 3);

    printEmps(arr, 3);
    // printf("%d", s(5));
```

```
    // printf("%d", s(3+2));
    return 0;
}
```

```
3
Enter name
Jenish
Enter salary
27000

id-> 1  Name-> Arjun  Salary->25000

id-> 2  Name-> Henil  Salary->30000

id-> 3  Name-> Jenish  Salary->27000
```

Q3)

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

typedef struct Admin{
    int id;
    char name[30];
    int salary;
    int allowance;
} Admin;

void storeStrct(Admin* emp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter id\n");
        scanf("%d", &emp[i].id);
        printf("Enter name\n");
        scanf("%s", emp[i].name);
        printf("Enter salary\n");
        scanf("%d", &emp[i].salary);
        printf("Enter allowance\n");
        scanf("%d", &emp[i].allowance);
    }
}
void printStrct(Admin* emp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\nid-> %d  Name-> %s  Salary->%d Allowance->%d\n", emp[i].id,
emp[i].name, emp[i].salary, emp[i].allowance);
    }
}


int main(){
    struct Admin arr[2];
```

```
    storeStrct(arr, 2);
    printStrct(arr, 2);
    return 0;
}
```

```
Enter id
1
Enter name
Jack
Enter salary
28000
Enter allowance
2500
Enter id
2
Enter name
Mike
Enter salary
15000
Enter allowance
5000

id-> 1  Name-> Jack  Salary->28000 Allowance->2500

id-> 2  Name-> Mike  Salary->15000 Allowance->5000
```

Q4)

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

typedef struct HR{
    int id;
    char name[30];
    int salary;
    int commision;
} HR;

void storeStrct(HR* h, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter id\n");
        scanf("%d", &h[i].id);
        printf("Enter name\n");
        scanf("%s", h[i].name);
        printf("Enter salary\n");
        scanf("%d", &h[i].salary);
        printf("Enter Commision\n");
        scanf("%d", &h[i].commision);
    }
}
void printStrct(HR* h, int n){
    for (int i = 0; i < n; i++)
    {
```

```
        printf("\nid-> %d  Name-> %s  Salary->%d Commision->%d\n", h[i].id, h[i].name,
h[i].salary, h[i].commision);
    }
}


int main(){
    struct HR h[2];
    storeStrct(h, 2);
    printStrct(h, 2);
    return 0;
}
```

```
Enter id
1
Enter name
Prience
Enter salary
28000
Enter Commision
2500
Enter id
2
Enter name
Yug
Enter salary
35000
Enter Commision
6000

id-> 1  Name-> Prience  Salary->28000 Commision->2500

id-> 2  Name-> Yug  Salary->35000 Commision->6000
```

Q5)

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

typedef struct SalesManager{
    int id;
    char name[30];
    int salary;
    int incentive;
    int target;
} SalesManager;

void storeStrct(SalesManager* h, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter id\n");
        scanf("%d", &h[i].id);
        printf("Enter name\n");
        scanf("%s", h[i].name);
        printf("Enter salary\n");
        scanf("%d", &h[i].salary);
        printf("Enter Incentive\n");
        scanf("%d", &h[i].incentive);
        printf("Enter Target\n");
        scanf("%d", &h[i].target);
    }
}
void printStrct(SalesManager* h, int n){
```

```
    for (int i = 0; i < n; i++)
    {
        printf("\nid-> %d  Name-> %s  Salary->%d Incentive->%d Target->%d\n", h[i].id,
h[i].name, h[i].salary, h[i].incentive, h[i].target);
    }
}

int main(){
    struct SalesManager smr[2];

    storeStrct(smr, 2);
    printStrct(smr, 2);
    return 0;
}
```

```
Enter salary
25000
Enter Incentive
5000
Enter Target
50
Enter id
2
Enter name
35000
Enter salary
100000
Enter Incentive
50000
Enter Target
500

id-> 1  Name-> Suraj  Salary->25000 Incentive->5000 Target->50

id-> 2  Name-> 35000  Salary->100000 Incentive->50000 Target->500
```

Q6)

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

typedef struct Date{
    int date;
    int month;
    int year;
} Date;

void storeStrct(Date* dt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter date\n");
        scanf("%d", &dt[i].date);
        printf("Enter month\n");
        scanf("%d", &dt[i].month);
        printf("Enter year\n");
        scanf("%d", &dt[i].year);
    }
}
void printStrct(Date* dt, int n){
    for (int i = 0; i < n; i++)
```

```
    {
        printf("\n%d/%d/%d\n", dt[i].date, dt[i].month, dt[i].year);
    }
}

int main(){
    struct Date dt[2];
    storeStrct(dt, 2);
    printStrct(dt, 2);

    return 0;
}
```

```
Enter date
22
Enter month
9
Enter year
2024
Enter date
15
Enter month
5
Enter year
2021

22/9/2024

15/5/2021
```

Q7)

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

typedef struct Time{
    int hr;
    int min;
    int sec;
} Time;

void storeStrct(Time* dt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter Hour\n");
        scanf("%d", &dt[i].hr);
        printf("Enter Minutes\n");
        scanf("%d", &dt[i].min);
        printf("Enter Seconds\n");
        scanf("%d", &dt[i].sec);
    }
}
void printStrct(Time* dt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\nHr-> %d Minutes->%d Sec->%d\n", dt[i].hr, dt[i].min, dt[i].sec);
    }
```

```
}

int main(){
    struct Time t[2];
    t[1].hr = 2;
    t[1].min = 20;
    t[1].sec = 23;
    storeStrct(t, 1);
    printStrct(t, 2);

    return 0;
}
```

```
Enter Hour
2
Enter Minutes
12
Enter Seconds
24

Hr-> 2 Minutes->12 Sec->24

Hr-> 2 Minutes->20 Sec->23
```

Q8)

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

typedef struct Distance{
    float feet;
    float inch;
} Distance;

void storeStrct(Distance* dt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter Feet\n");
        scanf("%f", &dt[i].feet);
        printf("Enter Inch\n");
        scanf("%f", &dt[i].inch);
    }
}
void printStrct(Distance* dt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\nLength is -> %.2f feets %.2f  inchs\n", dt[i].feet, dt[i].inch);
        printf("OR\n");
        printf("Length is -> %.2f' %.2f'", dt[i].feet, dt[i].inch);
    }
}

int main(){
    Distance t[2];
    t[1].feet = 2;
```

```
    t[1].inch = 20;

    storeStrct(t, 1);
    printStrct(t, 2);
    return 0;
}
```

```
Enter Feet
2
Enter Inch
10

Length is -> 2.00 feets 10.00  inchs
OR
Length is -> 2.00' 10.00''
Length is -> 2.00 feets 20.00  inchs
OR
Length is -> 2.00' 20.00''
```

Q9)

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

typedef struct Complex{
    int real;
    int img;
} Complex;

void storeStrct(Complex* cmp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter real num\n");
        scanf("%d", &cmp[i].real);
        printf("Enter imaginary num\n");
        scanf("%d", &cmp[i].img);
    }
}
void printStrct(Complex* cmp, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\n%d + %di\n", cmp[i].real, cmp[i].img);
    }
}

int main(){
    struct Complex cmp[2];
    cmp[1].real = 2;
    cmp[1].img = 20;

    storeStrct(cmp, 1);
    printStrct(cmp, 2);

    return 0;
}
```

```
Enter real num
5
Enter imaginary num
7

5 + 7i

2 + 20i
```

Q10)

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

typedef struct Product{
    int id;
    char name[30];
    int quantity;
    int price;
} Product;

void storeStrct(Product* prdt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("Enter product id\n");
        scanf("%d", &prdt[i].id);
        printf("Enter product name\n");
        scanf("%s", prdt[i].name);
        printf("Enter quantity\n");
        scanf("%d", &prdt[i].quantity);
        printf("Enter price\n");
        scanf("%d", &prdt[i].price);
    }
}
void printStrct(Product* prdt, int n){
    for (int i = 0; i < n; i++)
    {
        printf("\nid-> %d  Name-> %s  quantity->%d  price->%d totalAmount->%d rs.\n",
prdt[i].id, prdt[i].name, prdt[i].quantity, prdt[i].price, prdt[i].quantity *
prdt[i].price);
    }
}


int main(){
    Product p[2];

    p[1].id = 2;
    strcpy(p[1].name, "Milk");
    p[1].quantity = 5;
    p[1].price = 40;

    storeStrct(p, 1);
    printStrct(p, 2);
    return 0;
}
```

```
Enter product id
1
Enter product name
Mobile
Enter quantity
5
Enter price
17000

id-> 1  Name-> Mobile  quantity->5  price->17000 totalAmount->85000 rs.

id-> 2  Name-> Milk  quantity->5  price->40 totalAmount->200 rs.
```

------END------

Part 3: Pass by reference

Q1)

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

typedef struct Student{
    int rollNo;
    char name[30];
    int marks[3]; ///marks of two subjects
} Student;

void storeStudent(Student* s1){
    printf("Enter roll no\n");
    scanf("%d", &s1->rollNo);
    printf("Enter name no\n");
    scanf("%s", &s1->name);

    for (int i = 0; i < 3; i++)
    {
        printf("Enter marks for %d\n", i+1);
        scanf("%d", &s1->marks[i]);
    }
}

int main(){
    Student s1, s2;
    storeStudent(&s1);
    storeStudent(&s2);
```

```
    printf("\nRollNo-> %d  Name-> %s  Marks1->%d  Marks2->%d\n", s1.rollNo,
s1.name, s1.marks[0], s1.marks[1]);
    printf("\nRollNo-> %d  Name-> %s  Marks1->%d  Marks2->%d\n", s2.rollNo,
s2.name, s2.marks[0], s2.marks[1]);
    return 0;
}
```

```
Enter roll no
2
Enter name no
World
Enter marks for 1
90
Enter marks for 2
95
Enter marks for 3
78

RollNo-> 1  Name-> Hello  Marks1->85  Marks2->90

RollNo-> 2  Name-> World  Marks1->90  Marks2->95
```

Q2)

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

typedef struct Employee{
    int id;
    char name[30];
    int salary;
} Employee;

void storeEmps(Employee* emp){
        printf("Enter id\n");
        scanf("%d", &emp->id);
        printf("Enter name\n");
        scanf("%s", emp->name);
        printf("Enter salary\n");
        scanf("%d", &emp->salary);
}
void printEmps(Employee* emp){
        printf("\nid-> %d  Name-> %s  Salary->%d\n", emp->id, emp->name, emp-
>salary);
}


int main(){
    Employee e1, e2, e3;
    storeEmps(&e1);
```

```
    storeEmps(&e2);
    storeEmps(&e3);
    printEmps(&e1);
    printEmps(&e2);
    printEmps(&e3);

    return 0;
}
```

```
1
Enter name
Arjun
Enter salary
5000000
Enter id
2
Enter name
Rishi
Enter salary
2500000
Enter id
3
Enter name
rohit
Enter salary
4500000

id-> 1  Name-> Arjun  Salary->5000000

id-> 2  Name-> Rishi  Salary->2500000

id-> 3  Name-> rohit  Salary->4500000
```

Q3)

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

typedef struct HR{
    int id;
    char name[30];
    int salary;
    int commision;
} HR;

void storeStrct(HR* h){
        printf("Enter id\n");
        scanf("%d", &h->id);
        printf("Enter name\n");
        scanf("%s", h->name);
        printf("Enter salary\n");
        scanf("%d", &h->salary);
        printf("Enter Commision\n");
        scanf("%d", &h->commision);
}
void printStrct(HR* h){
        printf("\nid-> %d  Name-> %s  Salary->%d Commision->%d\n", h->id, h->name,
h->salary, h->commision);
}
```

```
int main(){
    struct HR h;
    storeStrct(&h);
    printStrct(&h);
    return 0;
}
```

```
 Enter id
 1
 Enter name
 Jack
 Enter salary
 35000
 Enter Commision
 5000

 id-> 1  Name-> Jack  Salary->35000 Commision->5000
```

Q4)

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

typedef struct Admin{
    int id;
    char name[30];
    int salary;
    int allowance;
} Admin;

void storeStrct(Admin* admn){
        printf("Enter id\n");
        scanf("%d", &admn->id);
        printf("Enter name\n");
        scanf("%s", admn->name);
        printf("Enter salary\n");
        scanf("%d", &admn->salary);
        printf("Enter allowance\n");
        scanf("%d", &admn->allowance);
}
void printStrct(Admin* admn){
        printf("\nid-> %d  Name-> %s  Salary->%d Allowance->%d\n", admn->id, admn-
>name, admn->salary, admn->allowance);
}


int main(){
    struct Admin a1;
```

```
    storeStrct(&a1);
    printStrct(&a1);
    return 0;
}
```

```
Enter id
1
Enter name
Yash
Enter salary
25000
Enter allowance
5000

id-> 1  Name-> Yash  Salary->25000 Allowance->5000
```

Q5)

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

typedef struct SalesManager{
    int id;
    char name[30];
    int salary;
    int incentive;
    int target;
} SalesManager;

void storeStrct(SalesManager* sm){
        printf("Enter id\n");
        scanf("%d", &sm->id);
        printf("Enter name\n");
        scanf("%s", sm->name);
        printf("Enter salary\n");
        scanf("%d", &sm->salary);
        printf("Enter Incentive\n");
        scanf("%d", &sm->incentive);
        printf("Enter Target\n");
        scanf("%d", &sm->target);
}
void printStrct(SalesManager* sm){
        printf("\nid-> %d  Name-> %s  Salary->%d Incentive->%d Target->%d\n", sm-
>id, sm->name, sm->salary, sm->incentive, sm->target);
}

int main(){
    struct SalesManager smr;

    storeStrct(&smr);
```

```
    printStrct(&smr);
    return 0;
}
```

```
Enter id
1
Enter name
Mike
Enter salary
25000
Enter Incentive
5000
Enter Target
215

id-> 1  Name-> Mike  Salary->25000 Incentive->5000 Target->215
```

Q6)

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

typedef struct Date
{
    int date;
    int month;
    int year;
} Date;

void storeStrct(Date *dt)
{
    printf("Enter date\n");
    scanf("%d", &dt->date);
    printf("Enter month\n");
    scanf("%d", &dt->month);
    printf("Enter year\n");
    scanf("%d", &dt->year);
}
void printStrct(Date *dt)
{
    printf("\n%d/%d/%d\n", dt->date, dt->month, dt->year);
}

int main()
{
    struct Date dt;
    storeStrct(&dt);
    printStrct(&dt);

    return 0;
```

```
}
```

```
Enter date
05
Enter month
12
Enter year
2024

5/12/2024
```

Q7)

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

typedef struct Time{
    int hr;
    int min;
    int sec;
} Time;

void storeStrct(Time* dt){
        printf("Enter Hour\n");
        scanf("%d", &dt->hr);
        printf("Enter Minutes\n");
        scanf("%d", &dt->min);
        printf("Enter Seconds\n");
        scanf("%d", &dt->sec);
}
void printStrct(Time* dt){
        printf("\nHr-> %d Minutes->%d Sec->%d\n", dt->hr, dt->min, dt->sec);
}

int main(){
    struct Time t;
    storeStrct(&t);
    printStrct(&t);

    return 0;
}
```

```
Enter Hour
2
Enter Minutes
23
Enter Seconds
12

Hr-> 2 Minutes->23 Sec->12
```

Q8)

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

typedef struct Distance
{
    float feet;
    float inch;
} Distance;

void storeStrct(Distance *dt)
{
    printf("Enter Feet\n");
    scanf("%f", &dt->feet);
    printf("Enter Inch\n");
    scanf("%f", &dt->inch);
}
void printStrct(Distance *dt)
{
    printf("\nLength is -> %.2f feets %.2f  inchs\n", dt->feet, dt->inch);
    printf("OR\n");
    printf("Length is -> %.2f' %.2f'", dt->feet, dt->inch);
}

int main()
{
    Distance t;

    storeStrct(&t);
    printStrct(&t);
    return 0;
}
```

```
Enter Feet
4
Enter Inch
11

Length is -> 4.00 feets 11.00  inchs
OR
Length is -> 4.00' 11.00''
```

Q9)

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

typedef struct Complex{
    int real;
    int img;
} Complex;

void storeStrct(Complex* cmp){
        printf("Enter real num\n");
        scanf("%d", &cmp->real);
        printf("Enter imaginary num\n");
        scanf("%d", &cmp->img);

}
void printStrct(Complex* cmp){
        printf("\n%d + %di\n", cmp->real, cmp->img);
}

int main(){
    struct Complex cmp;

    storeStrct(&cmp);
    printStrct(&cmp);

    return 0;
}
```

```
]633;CEnter real num
2
Enter imaginary num
3

2 + 3i
```

Q10)

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

```c
#include<string.h>

typedef struct Product{
    int id;
    char name[30];
    int quantity;
    int price;
} Product;

void storeStrct(Product* prdt){
        printf("Enter product id\n");
        scanf("%d", &prdt->id);
        printf("Enter product name\n");
        scanf("%s", prdt->name);
        printf("Enter quantity\n");
        scanf("%d", &prdt->quantity);
        printf("Enter price\n");
        scanf("%d", &prdt->price);
}
void printStrct(Product* prdt){
        printf("\nid-> %d  Name-> %s  quantity->%d  price->%d totalAmount->%d
rs.\n", prdt->id, prdt->name, prdt->quantity, prdt->price, prdt->quantity * prdt-
>price);
}


int main(){
    Product p;

    storeStrct(&p);
    printStrct(&p);
    return 0;
}
```

```
Enter product id
1
Enter product name
Pendrive
Enter quantity
5
Enter price
450

id-> 1  Name-> Pendrive  quantity->5  price->450 totalAmount->2250 rs.
```

Assignment 14 continue:

Q1) Create a structure Book with data members as name, id, author, price. Accept the values of all these members from user and display them.

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

typedef struct Book{
    char name[30];
    int id;
    char author[30];
    double price;
} Book;

void fgetsInput(char* str, int size){
    int len = strlen(str);
    fflush(stdin);
    if(fgets(str, size, stdin)){
        if(len>0 && str[len-1]=='\n') str[len-1] = '\0';
    }
    fflush(stdin);
}

void displayAll(Book* books,int n){
    for (int i = 0; i < n; i++)
    {
        printf("Book Name -> %s", books[i].name);
        printf("Book id -> %d\n", books[i].id);
        printf("Author Name -> %s", books[i].author);
        printf("Book id -> %.2lf\n", books[i].price);
    }
}

int main(){
    Book books[3];
    for (int i = 0; i < 3; i++)
    {
        printf("---Book %d---\n", i+1);
        printf("Enter name of the book\n");
        fgetsInput(books[i].name, sizeof(books[i].name));
        printf("Enter id of book\n");
        scanf("%d", &books[i].id);
        printf("Enter name of the author\n");
        fgetsInput(books[i].author, sizeof(books[i].name));
        printf("Enter price of book\n");
        scanf("%lf", &books[i].price);
    }

    displayAll(books, 3);

    return 0;
}
```

```
Enter name of the author
virat kohli
Enter price of book
1500
Book Name -> c plus plus
Book id -> 1
Author Name -> arjun patel
Book id -> 250.00
Book Name -> python
Book id -> 2
Author Name -> rohit sharma
Book id -> 550.00
Book Name -> java
Book id -> 3
Author Name -> virat kohli
Book id -> 1500.00
```

Q2)

```c
#include<stdio.h>

typedef struct Time{
    int hr;
    int min;
    char sec;
} Time;

void displayAll(Time* t,int n){
    for (int i = 0; i < n; i++)
    {
        printf("%d : %d : %d\n", t[i].hr, t[i].min, t[i].sec);
    }
}

int convertIntoSecs(Time t){
    int total = (t.hr*3600) + (t.min*60) + (t.sec);
    return total;
}

void add(Time t1, Time t2){
    int totalSecs = convertIntoSecs(t1) + convertIntoSecs(t2);
    int totalHrs =  totalSecs/3600;
    int hrRem = totalSecs%3600;
    int totalMin =  hrRem/60;
    int minRem = hrRem%60;
    int totalSec =  minRem%60;

    printf("%d : %d : %d\n", totalHrs, totalMin, totalSec);
}

int main(){
    Time t[2];
    for (int i = 0; i < 2; i++)
    {
```

```
        printf("---Time %d---\n", i+1);
        printf("Enter hrs\n");
        scanf("%d", &t[i].hr);
        printf("Enter minutes\n");
        scanf("%d", &t[i].min);
        printf("Enter secs\n");
        scanf("%d", &t[i].sec);
    }

    displayAll(t, 2);

    printf("Addition of two time\n");
    add(t[0], t[1]);

    return 0;
}
```

```
---Time 1---
Enter hrs
2
Enter minutes
50
Enter secs
40
---Time 2---
Enter hrs
2
Enter minutes
40
Enter secs
30
2 : 50 : 40
2 : 40 : 30
Addition of two time
5 : 31 : 10
```

Q3) Write a program to create an array for 10 players. For each player store name, no. of matches played, runs, wickets takes.
a. Create function to Accept the information of each player.
b. Create function to display the information of all the players
c. Display the information of player who made maximum runs and the one who took maximum number of wickets.

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

/*
Write a program to create an array for 10 players. For each player store name, no. of matches
played, runs, wickets takes.
a. Create function to Accept the information of each player.
b. Create function to display the information of all the players
c. Display the information of player who made maximum runs and the one who took maximum number
of wickets.
*/
```

```c
typedef struct Players
{
    char name[30];
    int noOfMatches;
    int runs;
    int wickets;
} Players;

void fgetsInput(char *str, size_t size)
{
    fflush(stdin);
    if (fgets(str, size, stdin))
    {
        // Remove newline character from fgets
        int len = strlen(str);
        if (len > 0 && str[len - 1] == '\n')
        {
            str[len - 1] = '\0';
        }
    }
}

void displayAll(Players *p, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("Name -> %s | ", p[i].name);
        printf("Matchs Played -> %d | ", p[i].noOfMatches);
        printf("Runs -> %d | ", p[i].runs);
        printf("Wickets -> %d\n", p[i].wickets);
    }
}
void displayByIndex(Players *p, int i)
{
    printf("Name -> %s | ", p[i].name);
    printf("Matchs Played -> %d | ", p[i].noOfMatches);
    printf("Runs -> %d | ", p[i].runs);
    printf("Wickets -> %d\n", p[i].wickets);
}

void storeStruct(Players *p, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("---Players %d---\n", i + 1);
        printf("Enter name of the Players\n");
        fgetsInput(p[i].name, sizeof(p[i].name));
        printf("Enter No of matching played\n");
        scanf("%d", &p[i].noOfMatches);
        printf("Enter runs score by player\n");
        scanf("%d", &p[i].runs);
        printf("Enter wickets taken by player\n");
        scanf("%d", &p[i].wickets);
    }
}

void displayTableToppers(Players *p, int n)
{
    int maxRuns = p[0].runs, maxWicket = p[0].wickets;
    int maxRunsIndex = 0, maxWicketIndex = 0;
    for (int i = 0; i < n; i++)
```

```
    {
        if (p[i].runs > maxRuns)
        {
            maxRuns = p[i].runs;
            maxRunsIndex = i;
        }
        if (p[i].wickets > maxWicket)
        {
            maxRuns = p[i].wickets;
            maxWicketIndex = i;
        }
    }

    printf("Player who score maximum runs\n");
    displayByIndex(p, maxRunsIndex);
    printf("Player who taken maximum wickets\n");
    displayByIndex(p, maxWicketIndex);
}

int main()
{
    Players p[3];
    storeStruct(p, 3);

    displayAll(p, 3);
    displayTableToppers(p, 3);

    return 0;
}
```

```
Name -> Arjun | Matchs Played -> 24 | Runs -> 1200 | Wickets -> 25
Name -> Rohit | Matchs Played -> 45 | Runs -> 121 | Wickets -> 255
Name -> Rishi | Matchs Played -> 25 | Runs -> 125 | Wickets -> 20
Player who score maximum runs
Name -> Arjun | Matchs Played -> 24 | Runs -> 1200 | Wickets -> 25
Player who taken maximum wickets
Name -> Rohit | Matchs Played -> 45 | Runs -> 121 | Wickets -> 255
```

Q4) Point of Sale System: Build a simple point of sale system using structures to represent products with attributes like name, price, and quantity. Allow users to add items to a cart and calculate the total cost.

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

/*
Point of Sale System: Build a simple point of sale system using structures to represent products with
attributes like name, price, and quantity. Allow users to add items to a cart and calculate the total cost.
*/

typedef struct Products
{
    int id;
    char name[30];
    double price;
} Products;

typedef struct CartItems
{
    Products p1;
```

```c
    int quantity;
} CartItems;

void fgetsInput(char *str, size_t size)
{
    fflush(stdin);
    if (fgets(str, size, stdin))
    {
        // Remove newline character from fgets
        int len = strlen(str);
        if (len > 0 && str[len - 1] == '\n')
        {
            str[len - 1] = '\0';
        }
    }
}

void displayAll(Products *p, int n)
{
    printf("---------------------------------\n");
    printf("|  Id  |    Product     | Price  |\n");
    printf("|------|----------------|--------|\n");
    for (int i = 0; i < n; i++)
    {
        int len = strlen(p[i].name);
        printf("|  %d   ", p[i].id);
        printf("| %s", p[i].name);
        for (int i = 1; i <= 15 - len; i++)
        {
            printf(" ");
        }
        i == 3 && printf("| %.2lf |\n", p[i].price);
        i != 3 && printf("| %.2lf  |\n", p[i].price);
    }
    printf("---------------------------------\n");
}
// void displayByIndex(Products *p, int i)
// {
//     printf("Name -> %s | ", p[i].name);
//     printf("Matchs Played -> %d | ", p[i].noOfMatches);
//     printf("Runs -> %d | ", p[i].runs);
//     printf("Wickets -> %d\n", p[i].wickets);
// }

// void storeStruct(Players *p, int n)
// {
//     for (int i = 0; i < n; i++)
//     {
//         printf("---Players %d---\n", i + 1);
//         printf("Enter name of the Players\n");
//         fgetsInput(p[i].name, sizeof(p[i].name));
//         printf("Enter No of matching played\n");
//         scanf("%d", &p[i].noOfMatches);
//         printf("Enter runs score by player\n");
//         scanf("%d", &p[i].runs);
//         printf("Enter wickets taken by player\n");
//         scanf("%d", &p[i].wickets);
//     }
// }

void generateAndDisplayBill(CartItems *c, int* n)
{
    double totalAmount = 0;
    printf("-----------------------------------------------------\n");
    printf("|  Id  |    Product     | Price  | Quantity |  Total  |\n");
    printf("|------|----------------|--------|----------|---------|\n");
```

```c
    for (int i = 0; i < *n; i++)
    {
        int len = strlen(c[i].p1.name);
        printf("|  %d   ", i+1);
        printf("| %s", c[i].p1.name);
        for (int i = 1; i <= 15 - len; i++)
        {
            printf(" ");
        }
        // printf("| %.2lf  \n", c[i].p1.price);
        c[i].p1.id == 3 && printf("| %.2lf ", c[i].p1.price);
        c[i].p1.id != 3 && printf("| %.2lf  ", c[i].p1.price);
        printf("|  %d      ", c[i].quantity);
        printf("| %.2lf |\n", c[i].quantity * c[i].p1.price);
        totalAmount += c[i].quantity * c[i].p1.price;

    }
    printf("-----------------------------------------------------\n");
    printf("                    Total Amount to pay ---> %.2lf\n\n\n", totalAmount);
}

void addToCart(CartItems *c, Products *p, int *i)
{
    int choice=1, pid;
    while (choice)
    {
        printf("Enter Product id of product want to add to cart\n");
        scanf("%d", &pid);
        //adding product at index sent from main
        c[*i].p1 = p[pid - 1];
        printf("Enter quantity\n");
        printf("%d %d  \n", *i , c[*i].quantity);
        (*i)++;

        printf("\nPress 1 to Add more items\n");
        printf("Press 0 to generate bill\n");
        scanf("%d", &choice);
        if(choice) displayAll(p,5);
    }
    generateAndDisplayBill(c, &(*i));
}

void hardCodedProducts(Products *p)
{
    p[0].id = 1;
    strcpy(p[0].name, "Milk");
    p[0].price = 30;

    p[1].id = 2;
    strcpy(p[1].name, "Pen");
    p[1].price = 10;

    p[2].id = 3;
    strcpy(p[2].name, "Chips");
    p[2].price = 20;

    p[3].id = 4;
    strcpy(p[3].name, "Washing Powder");
    p[3].price = 150;

    p[4].id = 5;
    strcpy(p[4].name, "Soap");
    p[4].price = 20;
}

int main()
```

```c
{
    int cartItemIndex = 0;
    Products p[5];
    CartItems c[50];
    // storeStruct(p, 3);
    hardCodedProducts(p);
    displayAll(p, 5);

    addToCart(c, p, &cartItemIndex);
    // displayTableToppers(p, 3);

    return 0;
}
```

```
Press 1 to Add more items
Press 0 to generate bill
2
---------------------------------
|  Id  |    Product     | Price  |
|------|----------------|--------|
|  1   | Milk           | 30.00  |
|  2   | Pen            | 10.00  |
|  3   | Chips          | 20.00  |
|  4   | Washing Powder | 150.00 |
|  5   | Soap           | 20.00  |
---------------------------------
Enter Product id of product want to add to cart
2
Enter quantity
4
3 4

Press 1 to Add more items
Press 0 to generate bill
0
-----------------------------------------------------
|  Id  |    Product     | Price  | Quantity |  Total  |
|------|----------------|--------|----------|---------|
|  1   | Milk           | 30.00  |  5       | 150.00 |
|  2   | Washing Powder | 150.00 |  1       | 150.00 |
|  3   | Soap           | 20.00  |  2       | 40.00 |
|  4   | Pen            | 10.00  |  4       | 40.00 |
-----------------------------------------------------
                    Total Amount to pay ---> 380.00
```

Q5) Movie Database: Create a program that uses structures to manage a movie database with details like title, director, release year, and genre. Allow users to add, search for, and update movie records.

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

```c
/*
Movie Database: Create a program that uses structures to manage a movie
database with details like title, director, release year, and genre. Allow
users to add, search for, and update movie records.
*/

typedef struct Movie
{
    int id;
    char title[30];
    char director[30];
    int releaseYear;
    char genre[30];
} Movie;

void hardCoded(Movie *mrr)
{
    mrr[0].id = 1;
    strcpy(mrr[0].title, "Animal");
    strcpy(mrr[0].director, "Arjun Reddy Vanga");
    mrr[0].releaseYear = 2024;
    strcpy(mrr[0].genre, "Action Crime");

    mrr[1].id = 2;
    strcpy(mrr[1].title, "3 Idiots");
    strcpy(mrr[1].director, "Vidhu Vinod Chopra");
    mrr[1].releaseYear = 2014;
    strcpy(mrr[1].genre, "Drama");

    mrr[2].id = 3;
    strcpy(mrr[2].title, "Stree");
    strcpy(mrr[2].director, "Maddock Production");
    mrr[2].releaseYear = 2023;
    strcpy(mrr[2].genre, "Horror Comedy");
}

void displayMovies(Movie *mrr, int *index)
{
    printf("-----------------------------------------------------------------
-----------------------------\n");
    printf("| %-3s | %-20s | %-20s | %-12s | %-20s |\n", "Id", "Title",
"Director", "Release Year", "Genre");
    printf("-----------------------------------------------------------------
-----------------------------\n");
    for (int i = 0; i < *index; i++)
```

```
    {
        printf("| %-3d | %-20s | %-20s | %-12d | %-20s |\n", mrr[i].id,
mrr[i].title, mrr[i].director, mrr[i].releaseYear, mrr[i].genre);
        printf("-------------------------------------------------------------
--------------------------------\n");
    }
}

int addMovie(Movie *mrr, int *index)
{
    static int staticId = 4;
    mrr[*index].id = staticId;
    printf("Assigned id --> %d\n", staticId);
    printf("Enter Movie Title\n");
    fflush(stdin);
    fgets(mrr[*index].title, sizeof(mrr[*index].title), stdin);
    // Optional: Remove newline character
    mrr[*index].title[strcspn(mrr[*index].title, "\n")] = '\0';

    printf("Enter the name of director\n");
    fgets(mrr[*index].director, sizeof(mrr[*index].director), stdin);
    // Optional: Remove newline character
    mrr[*index].director[strcspn(mrr[*index].director, "\n")] = '\0';

    fflush(stdin);
    printf("Enter release year of movie\n");
    scanf("%d", &mrr[*index].releaseYear);
    fflush(stdin);

    printf("Enter the genre of movie\n");
    fgets(mrr[*index].genre, sizeof(mrr[*index].genre), stdin);
    // Optional: Remove newline character
    mrr[*index].genre[strcspn(mrr[*index].genre, "\n")] = '\0';

    staticId++;
    return mrr[*index].id;
}

int searchById(Movie *mrr, int *index, int tempId)
{
    for (int i = 0; i < *index; i++)
    {
        if (mrr[i].id = tempId)
            return i;
    }
}
```

```c
void searchByName(Movie *mrr, int *index)
{
    char str[30];
    Movie tempArr[10];
    int j = 0;
    fflush(stdin);
    printf("Enter name of Movie you want to search/update\n");
    fgets(str, sizeof(str), stdin);
    fflush(stdin);
    // Optional: Remove newline character
    str[strcspn(str, "\n")] = '\0';

    for (int i = 0; i < *index; i++)
    {
        if (strstr(strlwr(mrr[i].title), strlwr(str)))
        {
            tempArr[j] = mrr[i];
            j++;
        }
    }

    if (j)
        displayMovies(tempArr, &j);
    else
        printf("No result found!\n");
}

void updateMovie(Movie *mrr, int *index)
{
    int tempId;
    searchByName(mrr, index);
    printf("Enter id of movie u want to update\n");
    scanf("%d", &tempId);

    int res = searchById(mrr, index, tempId);
    char tempName[30];
    int choice;
    printf("1. Update movie title\n");
    printf("2. Update movie director\n");
    printf("3. Update movie release year\n");
    printf("4. Update movie genre\n");
    scanf("%d", &choice);
    fflush(stdin);
    switch (choice)
    {
```

```c
    case 1:
        printf("Enter Movie Title\n");
        fflush(stdin);
        fgets(tempName, sizeof(tempName), stdin);
        // Optional: Remove newline character
        tempName[strcspn(tempName, "\n")] = '\0';
        strcpy(mrr[res].title, tempName);
        break;

    case 2:
        printf("Enter new director name\n");
        fflush(stdin);
        fgets(tempName, sizeof(tempName), stdin);
        // Optional: Remove newline character
        tempName[strcspn(tempName, "\n")] = '\0';
        strcpy(mrr[res].director, tempName);
        break;

    case 3:
        int tempYr;
        printf("Enter new release year\n");
        scanf("%d", &tempYr);
        mrr[res].releaseYear = tempYr;
        break;

    case 4:
        printf("Enter new genre\n");
        fflush(stdin);
        fgets(tempName, sizeof(tempName), stdin);
        // Optional: Remove newline character
        tempName[strcspn(tempName, "\n")] = '\0';
        strcpy(mrr[res].genre, tempName);
        break;

    default:
        printf("Invalid Choice\n");
        break;
    }
}

int main()
{
    int mIndex = 3, choice;
    Movie mrr[5];
    hardCoded(mrr);
```

```c
    while (1)
    {
        printf("1. Add Movie\n");
        printf("2. Display All Movies\n");
        printf("3. Search Movie by Name\n");
        printf("4. Update Movie\n");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
        {
            int res = addMovie(mrr, &mIndex);
            if (res)
            {
                printf("Movie added successfully with id -> %d\n", res);
                mIndex++;
            }
            else
                printf("Error in adding book\n");
            break;
        }
        case 2:
        {
            displayMovies(mrr, &mIndex);
            break;
        }
        case 3:
        {
            searchByName(mrr, &mIndex);
            break;
        }
        case 4:
        {
            updateMovie(mrr, &mIndex);
            break;
        }

        default:
            break;
        }
    }

    return 0;
}
```

```
Enter id of movie u want to update
2
1. Update movie title
2. Update movie director
3. Update movie release year
4. Update movie genre
4
Enter new genre
friends
1. Add Movie
2. Display All Movies
3. Search Movie by Name
4. Update Movie
2
```

Id	Title	Director	Release Year	Genre
2	hello	Arjun Reddy Vanga	2024	friends
2	3 idiots	Vidhu Vinod Chopra	2014	Drama
3	stree	Maddock Production	2023	Horror Comedy

```
1. Add Movie
2. Display All Movies
3. Search Movie by Name
4. Update Movie
```