```c
#include<stdio.h>
#include<string.h>
static struct student
{
    char name[20];
    int roll;
    float mark;
};
void Addrecord(struct student s[],int count)
{
    printf("Enter name\n");
    scanf("%s",s[count].name);
    printf("Enter roll number\n");
    scanf("%d",&s[count].roll);
    printf("Enter marks\n");
    scanf("%f",&s[count].mark);
}
void display(struct student s[],int count)
{
    for(int i=0;i<count;i++)
    {
        printf("Student %d\n",i+1);
        printf("Name is : %s\n",s[i].name);
        printf("Roll is : %d\n",s[i].roll);
        printf("Marks is : %.2f\n",s[i].mark);
        printf("\n");
    }
}
void displayroll(int r,int count,struct student *s)
{
    for(int i=0;i<count;i++)
    {
        if(s[i].roll==r)
        {
            printf("Name is : %s\n",s[i].name);
            printf("Roll is : %d\n",s[i].roll);
            printf("Marks is : %.2f\n",s[i].mark);
        }
    }
}
```

```c
void average(int count , struct student *s)
{
    float sum=0;
    for(int i=0;i<count;i++)
    {
        sum+=s[i].mark;
    }
    printf("Average marks is %.2f\n",sum/count);
}
void main()
{
    struct student s[10];
    int c,count=0,r;
    do
    {

        printf("1.Add record\n");
        printf("2.Display all records\n");
        printf("3.Display record by roll number\n");
        printf("4.Display average marks of all students\n");
        printf("5.Exit\n");
        printf("Enter option\n");
        scanf("%d",&c);
        switch(c)
        {
            case 1:
            Addrecord(s,count);
            count++;
            printf("Student added successfully\n");
            break;
            case 2:
            display(s,count);
            break;
            case 3:printf("Enter the roll number\n");
            scanf("%d",&r);
            displayroll(r,count,s);
            break;
            case 4:
            average(count,s);
            break;
```

```
            case 5:printf("EXITING....\n");
            break;
            default:
            break;
        }
    } while (c!=5);

}
```

```
1.Add record
2.Display all records
3.Display record by roll number
4.Display average marks of all students
5.Exit
Enter option
1
Enter name
rahul
Enter roll number
1
Enter marks
30
Student added successfully
1.Add record
2.Display all records
3.Display record by roll number
4.Display average marks of all students
5.Exit
Enter option
1
Enter name
gokul
Enter roll number
2
Enter marks
60
Student added successfully
1.Add record
2.Display all records
3.Display record by roll number
4.Display average marks of all students
5.Exit
Enter option
2
Student 1
Name is : rahul
Roll is : 1
Marks is : 30.00
```

```c
/*write a program to print month and number of days*/
#include<stdio.h>
struct day
{
    int days;
    char month[3];
};
void main()
{
    struct day
d[12]={{31,"JAN"},{28,"FEB"},{31,"MAR"},{30,"APR"},{31,"MAY"},{30,"JUN"},{
31,"JUL"},{31,"AUG"},{30,"SEP"},
    {31,"OCT"},{30,"NOV"},{31,"DEC"}};
      for(int i=0;i<12;i++)
    {
        printf("%s : %d\n",d[i].month,d[i].days);
    }
}
```

```
PS D:\projects\qu
JAN : 31
FEB : 28
MAR : 31
APR : 30
MAY : 31
JUN : 30
JUL : 31
AUG : 31
SEP : 30
OCT : 31
NOV : 30
DEC : 31
PS D:\projects\qu
```

```c
#include <stdio.h>

int main()
{
struct ptrs{
    int *ptr1;
```

```
    int *ptr2;
};
struct ptrs pointers;
int i1=100,i2;
pointers.ptr1=&i1;
pointers.ptr2=&i2;
*pointers.ptr2=200;
printf("address of i1 :%p\n",pointers.ptr1);
printf("address of i2 :%p\n",pointers.ptr2);
printf("value of i1 :%d\n",*pointers.ptr1);
printf("value of i2 :%d\n",*pointers.ptr2);
}
```

```
PS D:\projects\quest\C> cd "d:\
address of i1 :0061FF14
address of i2 :0061FF10
value of i1 :100
value of i2 :200
PS D:\projects\quest\C>
```

```
/*Define a structure to store student information, including name, roll
number, and marks in three subjects.
Write a program to input data for 5 students and display the details along
with their average marks.*/
#include<stdio.h>
struct student
{
    char name[20];
    int roll;
    float m1;
    float m2;
    float m3;
};
void main()
{
    float avg;
    struct student st[5];
    printf("Enter details of five students\n");
    for(int i=0;i<5;i++)
    {
```

```
        printf("Enter name of student %d\n",i+1);
        scanf("%s",&st[i].name);
        printf("Enter roll number\n");
        scanf("%d",&st[i].roll);
        printf("Enter the marks for three subjects\n");
        scanf("%f %f %f",&st[i].m1,&st[i].m2,&st[i].m3);
    }

    printf("\nDetails of students\n");
    for(int i=0;i<5;i++)
    {
        printf("NAme of student %d is : %s\n",i+1,st[i].name);
        printf("Roll number of student %d is : %d\n",i+1,st[i].roll);
        printf("Marks for student %d is -Sub1 : %.2f  | Sub2 : %.2f | Sub3
: %.2f\n",i+1,st[i].m1,st[i].m2,st[i].m3);
        avg=(st[i].m1+st[i].m2+st[i].m3)/3;
        printf("Average mark of student %d is %.2f\n",avg);
        printf("\n");
    }
}
```

```
Details of students
NAme of student 1 is : rahul
Roll number of student 1 is : 1
Marks for student 1 is -Sub1 : 30.00  | Sub2 : 40.00 | Sub3 : 50.00
Average mark of student 0 is 30.00

NAme of student 2 is : gokul
Roll number of student 2 is : 2
Marks for student 2 is -Sub1 : 50.00  | Sub2 : 60.00 | Sub3 : 70.00
Average mark of student 0 is 50.00

NAme of student 3 is : kris
Roll number of student 3 is : 3
Marks for student 3 is -Sub1 : 56.00  | Sub2 : 73.00 | Sub3 : 99.00
Average mark of student 0 is 56.00

NAme of student 4 is : hari
Roll number of student 4 is : 4
Marks for student 4 is -Sub1 : 29.00  | Sub2 : 49.00 | Sub3 : 59.00
Average mark of student 1610612736 is 29.00

NAme of student 5 is : arun
Roll number of student 5 is : 5
Marks for student 5 is -Sub1 : 60.00  | Sub2 : 90.00 | Sub3 : 80.00
Average mark of student -1610612736 is 60.00
```

```c
/*Create a structure to store employee details like name, ID, salary, and
department.
Write a function to display the details of employees whose salary is above
a certain threshold.*/
#include<stdio.h>
struct employee
{
    char name[20];
    char id[5];
    float salary;
    char department[10];
};
void main()
{
    struct employee e[5];
    float threshold;
    printf("Enter details of employee\n");
```

```
    for(int i=0;i<5;i++)
    {
        printf("Enter employee name\n");
        scanf("%s",&e[i].name);
        printf("Enter employee id\n");
        scanf("%s",&e[i].id);
        printf("Enter the salay\n");
        scanf("%f",&e[i].salary);
        printf("Enter department\n");
        scanf("%s",&e[i].department);
    }
    printf("Enter threshold salary\n");
    scanf("%f",&threshold);
    printf("Employees above threshold are:\n");
    for(int i=0;i<5;i++)
    {
        if(e[i].salary>threshold)
        {
            printf("Name : %s\n",e[i].name);
            printf("ID : %s\n",e[i].id);
            printf("Salary : %.2f\n",e[i].salary);
            printf("Department : %s\n",e[i].department);
            printf("\n");
        }
    }
}
```

```
Employees above threshold are:
Name : kris
ID : e3
Salary : 45000.00
Department : R&D

Name : gopi
ID : e4
Salary : 54000.00
Department : cse

Name : rayan
ID : e5
Salary : 60000.00
Department : R&D

PS D:\projects\guest\C>
```

```c
/*Define a structure to represent a book with fields for title, author,
ISBN, and price.
Write a program to manage an inventory of books and allow searching by
title.*/
#include<stdio.h>
#include<string.h>
struct library
{
    char title[20];
    char author[20];
    char ISBN[5];
    float price;
};
void main()
{
   struct library
books[5]={{"book1","author1","ISB1",250},{"book2","author2","ISB2",450},

{"book3","author3","ISB3",500},{"book4","author4","ISB4",600},{"book5","au
thor5","ISB5",650}};
   char  t[20];
   printf("Enter title of book\n");
   scanf("%s",t);
```

```
   for(int i=0;i<5;i++)
   {
    if(strcmp(t,books[i].title)==0)
    {
        printf("Book found\n");
        printf("Book title : %s\n",books[i].title);
        printf("Book author : %s\n",books[i].author);
        printf("Book ISBN : %s\n",books[i].ISBN);
        printf("Book price %d\n",books[i].price);
    }
   }

}
```

```
PS D:\projects\quest\C> cd "d:\proje
Enter title of book
book2
Book found
Book title : book2
Book author : author2
Book ISBN : ISB2
Book price 0
PS D:\projects\quest\C>
```

```
/*Create a structure to represent a date with day, month, and year.
Write a function to validate if a given date is correct (consider leap
years).*/
#include<stdio.h>
struct date
{
    int day;
    int month;
    int year;
};
int valid(struct date d)
{
     if(d.month==1 ||d.month==3 ||d.month==5 ||d.month==7 ||d.month==8
||d.month==10 ||d.month==12 )
    {
```

```
        if(d.day>1 && d.day<=31);
        else
        return 0;
    }
    else  if(d.month==4 ||d.month==6 ||d.month==9 ||d.month==11)
    {
        if(d.day>1 && d.day<=30);
        else
       return 0;
    }
    else
    {
        if(d.year%400==0)
        {
            if(d.month==2)
            {
                if(d.day<=29);
                else
                return 0;
            }
        }
        else if(d.year%100==0)
        {
            if(d.month==2)
            {
                if(d.day<=28);
                else
                return 0;
            }
        }
        else if(d.year%4 == 0)
           {
            if(d.month==2)
            {
                if(d.day<=29);
                else
                return 0;
            }
        }
        else
```

```
        return 0;
    }
    return 1;
}
void main()
{
    struct date d;
    int flag;
    printf("Enter the date\n");
    scanf("%d %d %d",&d.day,&d.month,&d.year);
   flag = valid(d);
   if(flag==1)
   printf("Valid date\n");
   else
   printf("Invalid date");

}
```

```
PS D:\projects\quest\C> cd "d:\proj
Enter the date
29 03 2001
Valid date
PS D:\projects\quest\C> cd "d:\proj
Enter the date
31 02 2001
Invalid date
PS D:\projects\quest\C>
```

```
/*Define a structure to represent a complex number with real and imaginary
parts.
Implement functions to add, subtract, and multiply two complex numbers.*/
#include<stdio.h>
struct num
{
    int n;
    int  m;
};
void add( struct num n1,struct num n2)
{
int sum1,sum2;
```

```
sum1=n1.n+n2.n;
sum2=n1.m+n2.m;
printf("Sum of %d + i%d and %d +i%d : %d +
i%d\n",n1.m,n1.m,n2.n,n2.m,sum1,sum2);
}
void sub( struct num n1,struct num n2)
{
    int dif1,dif2;
    dif1=n1.n-n2.n;
    dif2=n1.m-n2.m;
    printf("Difference between %d + i%d and %d +i%d : %d +
i%d\n",n1.m,n1.m,n2.n,n2.m,dif1,dif2);

}
void mult( struct num n1,struct num n2)
{
    int prod1,prod2;
    prod1=n1.n*n2.n-(n1.m*n2.m);
    prod2=n1.n*n2.m+n1.m*n2.n;
    printf("Product of %d + i%d and %d + i%d : %d +
i%d",n1.n,n1.m,n2.n,n2.m,prod1,prod2);
}
void main()
{
    struct num n1,n2;
    printf("Enter the first complex number\n");
    scanf("%d %d",&n1.n,&n1.m);
    printf("Enter the second complex number\n");
    scanf("%d %d",&n2.n,&n2.m);
    add(n1,n2);
    sub(n1,n2);
    mult(n1,n2);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ;
Enter the first complex number
1 3
Enter the second complex number
5 3
Sum of 3 + i3 and 5 +i3 : 6 + i6
Difference between 3 + i3 and 5 +i3 : -4 + i0
Product of 1 + i3 and 5 + i3 : -4 + i18
PS D:\projects\quest\C>
```

```c
/*Design a structure to store information about a bank account, including
account number, account holder name, and balance.
Write a function to deposit and withdraw money, and display the updated
balance.*/
#include<stdio.h>
struct account
{
    char name[20];
    char num[10];
    float balance;
};
void withdraw(struct account *n,float amount)
{
    if(amount>n->balance)
    {
        printf("Insufficient balance\n");
    }
    else
    {
        n->balance=n->balance-amount;
    printf("Balance amount is %.2f\n",n->balance);
    }

}
void deposit(struct account *n,float amount)
{
    if(amount<0)
    printf("Invalid amount");
```

```
    else
    {n->balance=n->balance+amount;
    printf("Balance amount is %.2f\n",n->balance);
    }


}
void main()
{
    struct account n={"Rahul","B123",25000.45};
    int c;
    float amount;
    do
    {
        printf("1.Deposit,2.Withraw\n");
        scanf("%d",&c);
        switch(c)
        {
            case 1:printf("Enter amount\n");
            scanf("%f",&amount);
            deposit(&n,amount);
            break;
            case 2:printf("Enter amount\n");
            scanf("%f",&amount);
            withdraw(&n,amount);
            break;
            case 3:
            printf("EXITING.....\n");
            break;
            default:
            printf("Enter valid input\n");
            break;
        }
    }while(c!=3);

}
```

```
PS D:\projects\quest\C> cd "d:\proj
1.Deposit,2.Withraw
1
Enter amount
3000
Balance amount is 28000.45
1.Deposit,2.Withraw
2
Enter amount
4000
Balance amount is 24000.45
1.Deposit,2.Withraw
3
EXITING.....
PS D:\projects\quest\C>
```

```c
/*Create a structure for a car with fields like make, model, year, and
price.
Write a program to store details of multiple cars and print cars within a
specified price range.*/
#include<stdio.h>
struct car{
    char make[20];
    char model[20];
    int year;
    float price;
};
void main()
{
    struct car
c[5]={{"BMW","M3",2001,6000},{"Toyota","Supra",2002,8500},{"Nissan","skyli
ne",2004,7800},
    {"Mitsubishi","lancer-evo",2003,5800},{"Subaru","WRX",2004,4800}};
    int p;
    printf("Enter the price range\n");
    scanf("%d",&p);
    printf("Available Cars\n");
```

```
    for(int i=0;i<5;i++)
    {
        if(p>c[i].price)
        {
            printf("Make : %s\n",c[i].make);
            printf("Model : %s\n",c[i].model);
            printf("Year : %d\n",c[i].year);
            printf("Price : %f\n",c[i].price);
            printf("\n");
        }
    }
}
```

```
PS D:\projects\quest\C> cd "d:\projects\c
Enter the price range
6000
Available Cars
Make : Mitsubishi
Model : lancer-evo
Year : 2003
Price : 5800.000000

Make : Subaru
Model : WRX
Year : 2004
Price : 4800.000000

PS D:\projects\quest\C>
```

```
/*Define a structure for a library book with fields for title, author,
publication year, and status (issued or available).
Write a function to issue and return books based on their status.*/
#include<stdio.h>
struct library
{
    char title[20];
    char author[20];
    int year;
    int status;
```

```
};
void process(struct library *book)
{
    if(book->status==1)
    book->status=0;
    else
    book->status=1;
}
void main()
{
    struct library
books[5]={{"Book1","Author1",2001,1},{"Book2","Author2",2012,0},

{"Book3","Author3",2002,1},{"Book4","Author4",2007,0},{"Book5","Author5",2
011,0}};
    for(int i=0;i<5;i++)
    {
        if(books[i].status==1)
        {
            printf("Book is issued \n");
            process(&books[i]);
            printf("Book %s returned\n",books[i].title);
            printf("\n");

        }
        else
        {
            printf("Book is available \n");
            process(&books[i]);
            printf("Issuing book %s\n",books[i].title);
            printf("\n");
        }

    }

}
```

```
PS D:\projects\quest\C> cd "d
Book is issued
Book Book1 returned

Book is available
Issuing book Book2

Book is issued
Book Book3 returned

Book is available
Issuing book Book4

Book is available
Issuing book Book5

PS D:\projects\quest\C>
```

```c
/*Create a structure to store a student's name, roll number, and an array
of grades.
Write a program to calculate and display the highest, lowest, and average
grade for each student*/
#include<stdio.h>
struct students
{
    char name[20];
    int roll;
    int grade[5];
};
void highest(int *ar,int n)
{
    int high=0;
    for(int i=0;i<5;i++)
    {
        if(ar[i]>high)
        high=ar[i];
    }
    printf("Highest grade is %d\n",high);
```

```
}
void lowest(int *ar,int n)
{
    int low=100;
    for(int i=0;i<5;i++)
    {
        if(ar[i]>low)
        low=ar[i];
    }
    printf("Lowest grade is %d\n",low);
}
void average(int *ar,int n)
{
    int sum=0;
    for(int i=0;i<5;i++)
    {
        sum+=ar[i];
    }
    printf("Average grade is %d\n",sum/5);
}

void main()
{
    struct students
set[5]={{"Rahul",1,{50,60,70,80,90}},{"Gokul",2,{78,62,71,80,43}},{"Kevin"
,3,{81,70,28,36,54}},{"Kris",4,{72,34,87,57,94}},{"Rohit",5,{63,47,50,18,7
9}}};
    for(int i=0;i<5;i++)
    {
        printf("Name : %s\n",set[i].name);
        printf("Roll : %d\n",set[i].roll);
        printf("Grades : ");
        for(int j=0;j<5;j++)
        {
            printf("%d ",set[i].grade[j]);
        }
        printf("\n");
        highest(set[i].grade,5);
        lowest(set[i].grade,5);
        average(set[i].grade,5);
```

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

```
Name : Rahul
Roll : 1
Grades : 50 60 70 80 90
Highest grade is 90
Lowest grade is 100
Average grade is 70

Name : Gokul
Roll : 2
Grades : 78 62 71 80 43
Highest grade is 80
Lowest grade is 100
Average grade is 66

Name : Kevin
Roll : 3
Grades : 81 70 28 36 54
Highest grade is 81
Lowest grade is 100
Average grade is 53

Name : Kris
Roll : 4
Grades : 72 34 87 57 94
Highest grade is 94
Lowest grade is 100
Average grade is 68

Name : Rohit
Roll : 5
Grades : 63 47 50 18 79
Highest grade is 79
Lowest grade is 100
Average grade is 51

PS D:\projects\quest\C>
```

```c
/*Define a structure to represent a product with fields for product ID,
name, quantity, and price.
Write a program to update the quantity of products after a sale and
calculate the total sales value.*/
#include <stdio.h>
#include <string.h>
struct product {
    int id;
    char name[50];
    int quantity;
    float price;
};
void processSale(struct product *p, int saleQuantity, float *totalSales) {
    if (saleQuantity > p->quantity) {
        printf("Insufficient stock for product '%s'. Available quantity:
%d\n", p->name, p->quantity);
    } else {
        p->quantity -= saleQuantity;
        float saleValue = saleQuantity * p->price;
        *totalSales += saleValue;
        printf("Sale processed for product '%s'. Sale value: %.2f\n",
p->name, saleValue);
    }
}
void displayProduct(struct product p) {
    printf("Product ID: %d\n", p.id);
    printf("Name: %s\n", p.name);
    printf("Quantity: %d\n", p.quantity);
    printf("Price: %.2f\n", p.price);
    printf("--------------------------\n");
}
void main() {
    int n=5, saleQuantity, choice;
    float totalSales = 0;
    struct product
products[5]={{1,"carrot",50,2},{2,"book",60,10},{3,"pen",100,5},{4,"ruler"
,50,3},{5,"umbrella",70,20}};
    do {
        printf("\n--- Sales Menu ---\n");
        printf("1. Process Sale\n");
```

```
        printf("2. Display Products\n");
        printf("3. Display Total Sales\n");
        printf("4. Exit\n");
        printf("Enter your choice:\n");
        scanf("%d", &choice);
        switch (choice) {
            case 1: {
                int productId;
                printf("Enter product ID for sale: ");
                scanf("%d", &productId);
                int found = 0;
                for (int i = 0; i < n; i++) {
                    if (products[i].id == productId) {
                        printf("Enter sale quantity: ");
                        scanf("%d", &saleQuantity);
                        processSale(&products[i], saleQuantity,
&totalSales);
                        found = 1;
                        break;
                    }
                }
                if (!found) {
                    printf("Product with ID %d not found.\n", productId);
                }
                break;
            }
            case 2:
                printf("\n--- Product Details ---\n");
                for (int i = 0; i < n; i++) {
                    displayProduct(products[i]);
                }
                break;

            case 3:
                printf("Total Sales Value: %.2f\n", totalSales);
                break;

            case 4:
                printf("Exiting...\n");
                break;
```

```
            default:
                printf("Invalid choice! Please try again.\n");
        }
    } while (choice != 4);
}
```

```
--- Sales Menu ---
1. Process Sale
2. Display Products
3. Display Total Sales
4. Exit
Enter your choice:
1
Enter product ID for sale: 1
Enter sale quantity: 20
Sale processed for product 'carrot'. Sale value: 40.00

--- Sales Menu ---
1. Process Sale
2. Display Products
3. Display Total Sales
4. Exit
Enter your choice:
2

--- Product Details ---
Product ID: 1
Name: carrot
Quantity: 30
Price: 2.00
--------------------------
Product ID: 2
Name: book
Quantity: 60
Price: 10.00
--------------------------
Product ID: 3
Name: pen
Quantity: 100
Price: 5.00
--------------------------
Product ID: 4
Name: ruler
Quantity: 50
Price: 3.00
```

```
Quantity: 100
Price: 5.00
-------------------------
Product ID: 4
Name: ruler
Quantity: 50
Price: 3.00
-------------------------
Product ID: 5
Name: umbrella
Quantity: 70
Price: 20.00
-------------------------

--- Sales Menu ---
1. Process Sale
2. Display Products
3. Display Total Sales
4. Exit
Enter your choice:
3
Total Sales Value: 40.00

--- Sales Menu ---
1. Process Sale
2. Display Products
3. Display Total Sales
4. Exit
Enter your choice:
4
Exiting...
PS D:\projects\quest\C>
```

```c
/*Define a structure for a point in 2D space (x, y).
Write a function to calculate the distance between two points.*/
#include<stdio.h>
#include<math.h>
```

```c
struct point
{
    int x;
    int y;
};
void main()
{
    struct point p1,p2;
    int d1,d2,d;
    printf("Enter the first point\n");
    scanf("%d %d",&p1.x,&p1.y);
    printf("Enter the second point\n");
    scanf("%d %d",&p2.x,&p2.y);
    d1=(p2.x-p1.x);
    d2=(p2.y-p1.y);
    d=d1*d1 + d2*d2;
    printf("Distance is %d",d);
}
```

```
Enter the first point
2 3
Enter the second point
5 7
Distance is 25
PS D:\projects\quest\C>
```

```c
/*Create a structure for a rectangle with length and width.
Write functions to calculate the area and perimeter of the rectangle.*/
#include<stdio.h>
struct rectangle{
    int l;
    int w;
};
void main()
{
    struct rectangle r;
    printf("Enter the length and width\n");
    scanf("%d %d",&r.l,&r.w);
    printf("Area is %d\n",r.l*r.w);
    printf("Perimeter is %d\n",2*(r.l+r.w));
```

```
}
```

```
PS D:\projects\quest\C> cd "d:\projects\
Enter the length and width
4 5
Area is 20
Perimeter is 18
PS D:\projects\quest\C>
```

```c
/*Define a structure to store details of a movie, including title,
director, release year, and rating.
Write a program to sort movies by their rating.*/
#include<stdio.h>
struct movie
{
    char title[20];
    char director[20];
    int year;
    int rating;
};
void sort(struct movie m[], int size) {
    struct movie temp;
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (m[i].rating > m[j].rating) {
                temp = m[i];
                m[i] = m[j];
                m[j] = temp;
            }
        }
    }
    printf("\nSorted list by rating:\n");
    for (int i = 0; i < size; i++) {
        printf("Title: %s\n", m[i].title);
        printf("Director: %s\n", m[i].director);
        printf("Year: %d\n", m[i].year);
        printf("Rating: %d\n", m[i].rating);
```

```
    }
}

void main() {
    struct movie m[5] = {
        {"Movie1", "Director1", 2001, 2},
        {"Movie2", "Director2", 2002, 4},
        {"Movie3", "Director3", 2007, 3},
        {"Movie4", "Director4", 2009, 5},
        {"Movie5", "Director5", 2007, 1}
    };
    printf("The movie details are:\n");
    for (int i = 0; i < 5; i++) {
        printf("Title: %s\n", m[i].title);
        printf("Director: %s\n", m[i].director);
        printf("Year: %d\n", m[i].year);
        printf("Rating: %d\n", m[i].rating);

    }
    printf("\n");
    sort(m, 5);
}
```

```
Sorted list by rating:
Title: Movie5
Director: Director5
Year: 2007
Rating: 1
Title: Movie1
Director: Director1
Year: 2001
Rating: 2
Title: Movie3
Director: Director3
Year: 2007
Rating: 3
Title: Movie2
Director: Director2
Year: 2002
Rating: 4
Title: Movie4
Director: Director4
Year: 2009
Rating: 5
PS D:\projects\quest\C>
```

```c
/*Create a structure to store daily weather data, including date,
temperature, and humidity.
Write a program to find the day with the highest temperature.*/
#include<stdio.h>
struct Weather {
    char date[15];
    float temperature;
    float humidity;
};
void main() {
    int n=5,max = 0;
```

```
    struct Weather
data[5]={{"8-1-2025",23.5,12},{"9-1-2025",27.2,7},{"10-1-2025",29.5,6},{"1
1-1-2025",34.5,2},{"12-1-2025",27.5,21}};
    for (int i = 1; i < n; i++) {
        if (data[i].temperature > data[max].temperature) {
            max = i;
        }
    }
    printf("\nDay with the highest temperature:\n");
    printf("Date: %s\n", data[max].date);
    printf("Temperature: %.2f\n", data[max].temperature);
    printf("Humidity: %.2f\n", data[max].humidity);
}
```

```
Day with the highest temperature:
Date: 11-1-2025
Temperature: 34.50
Humidity: 2.00
PS D:\projects\quest\C>
```

```
/*Define a structure for a fraction with numerator and denominator.
Write functions to add, subtract, multiply, and divide two fractions.*/
#include <stdio.h>
struct Fraction {
    int numerator;
    int denominator;
};
struct Fraction add(struct Fraction f1, struct Fraction f2) {
    struct Fraction result;
    result.numerator = f1.numerator * f2.denominator + f2.numerator *
f1.denominator;
    result.denominator = f1.denominator * f2.denominator;
    return result;
}
struct Fraction subtract(struct Fraction f1, struct Fraction f2) {
    struct Fraction result;
    result.numerator = f1.numerator * f2.denominator - f2.numerator *
f1.denominator;
```

```
    result.denominator = f1.denominator * f2.denominator;
    return result;
}
struct Fraction multiply(struct Fraction f1, struct Fraction f2) {
    struct Fraction result;
    result.numerator = f1.numerator * f2.numerator;
    result.denominator = f1.denominator * f2.denominator;
    return result;
}
struct Fraction divide(struct Fraction f1, struct Fraction f2) {
    struct Fraction result;
    result.numerator = f1.numerator * f2.denominator;
    result.denominator = f1.denominator * f2.numerator;
    return result;
}
void main() {
    struct Fraction f1, f2, result;

    printf("Enter numerator and denominator of first fraction: ");
    scanf("%d %d", &f1.numerator, &f1.denominator);

    printf("Enter numerator and denominator of second fraction: ");
    scanf("%d %d", &f2.numerator, &f2.denominator);

    if (f1.denominator == 0 || f2.denominator == 0) {
        printf("Error: Denominator cannot be zero.\n");
        return;
    }

    printf("\nAddition: ");
    result = add(f1, f2);
    printf("%d/%d\n", result.numerator, result.denominator);
    printf("Subtraction: ");
    result = subtract(f1, f2);
    printf("%d/%d\n", result.numerator, result.denominator);
    printf("Multiplication: ");
    result = multiply(f1, f2);
    printf("%d/%d\n", result.numerator, result.denominator);
    printf("Division: ");
    result = divide(f1, f2);
```

```
    printf("%d/%d\n", result.numerator, result.denominator);
}
```

```
Enter numerator and denominator of first fraction: 2 5
Enter numerator and denominator of second fraction: 3 7

Addition: 29/35
Subtraction: -1/35
Multiplication: 6/35
Division: 14/15
PS D:\projects\quest\C>
```

```c
/*Create a structure to represent a laptop with fields for brand, model,
processor, RAM, and price.
Write a program to list laptops within a specific price range.*/
#include <stdio.h>
#include <string.h>

struct Laptop {
    char brand[20];
    char model[20];
    char processor[20];
    int RAM;
    float price;
};

void listLaptops(struct Laptop laptops[], int size, float minPrice, float
maxPrice) {
    int found = 0;
    printf("Laptops within the price range $%.2f - $%.2f:\n", minPrice,
maxPrice);
    for (int i = 0; i < size; i++) {
        if (laptops[i].price >= minPrice && laptops[i].price <= maxPrice)
{
            printf("Brand: %s\n", laptops[i].brand);
            printf("Model: %s\n", laptops[i].model);
            printf("Processor: %s\n", laptops[i].processor);
```

```
            printf("RAM: %dGB\n", laptops[i].RAM);
            printf("Price: %.2f\n\n", laptops[i].price);
            found = 1;
        }
    }
    if (!found) {
        printf("No laptops found within this price range.\n");
    }
}
void main() {
    struct Laptop laptops[5] = {
        {"Dell", "Inspiron", "Intel i5", 8, 600.00},
        {"HP", "Pavilion", "Intel i7", 16, 900.00},
        {"Apple", "MacBook Air", "M1", 8, 999.00},
        {"Lenovo", "ThinkPad", "AMD Ryzen 5", 16, 700.00},
        {"Asus", "VivoBook", "Intel i3", 4, 400.00}
    };
    float minPrice, maxPrice;
    printf("Enter minimum price: ");
    scanf("%f", &minPrice);
    printf("Enter maximum price: ");
    scanf("%f", &maxPrice);
    listLaptops(laptops, 5, minPrice, maxPrice);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($
Enter minimum price: 300
Enter maximum price: 700
Laptops within the price range $300.00 - $700.00:
Brand: Dell
Model: Inspiron
Processor: Intel i5
RAM: 8GB
Price: 600.00

Brand: Lenovo
Model: ThinkPad
Processor: AMD Ryzen 5
RAM: 16GB
Price: 700.00

Brand: Asus
Model: VivoBook
Processor: Intel i3
RAM: 4GB
Price: 400.00

PS D:\projects\quest\C>
```

```c
/*Define a structure to store attendance data, including student ID, total
classes, and classes attended.
Write a program to calculate and display the attendance percentage for
each student.*/
#include <stdio.h>
struct Attendance {
    int studentID;
    int totalClasses;
    int classesAttended;
};
void calculateAttendance(struct Attendance students[], int size)
 {
    printf("Attendance Report:\n");
    printf("Student ID\tTotal Classes\tClasses Attended\tPercentage\n");
```

```
    for (int i = 0; i < size; i++) {
        float percentage = ((float)students[i].classesAttended /
students[i].totalClasses) * 100;
        printf("%d\t\t%d\t\t%d\t\t\t%.2f%%\n",
               students[i].studentID,
               students[i].totalClasses,
               students[i].classesAttended,
               percentage);
    }
}
void main()
 {
    int numStudents;
    printf("Enter the number of students: ");
    scanf("%d", &numStudents);
    struct Attendance students[numStudents];
    for (int i = 0; i < numStudents; i++)
    {
        printf("Enter data for student %d:\n", i + 1);
        printf("Student ID: ");
        scanf("%d", &students[i].studentID);
        printf("Total Classes: ");
        scanf("%d", &students[i].totalClasses);
        printf("Classes Attended: ");
        scanf("%d", &students[i].classesAttended);
    }
    calculateAttendance(students, numStudents);
}
```

```
Enter the number of students: 3
Enter data for student 1:
Student ID: 1
Total Classes: 5
Classes Attended: 3
Enter data for student 2:
Student ID: 2
Total Classes: 5
Classes Attended: 4
Enter data for student 3:
Student ID: 3
Total Classes: 5
Classes Attended: 1
Attendance Report:
Student ID      Total Classes   Classes Attended        Percentage
1               5               3                       60.00%
2               5               4                       80.00%
3               5               1                       20.00%
PS D:\projects\quest\C>
```

```c
/*Create a structure for a flight with fields for flight number,
departure, destination, and duration.
Write a program to display flights that are less than a specified
duration.*/
#include <stdio.h>
#include <string.h>
struct Flight {
    char flightNumber[10];
    char departure[20];
    char destination[20];
    float duration;
};
void displayFlights(struct Flight flights[], int size, float maxDuration)
{
    printf("\nFlights with duration less than %.2f hours:\n",
maxDuration);
    printf("Flight No\tDeparture\tDestination\tDuration\n");
    for (int i = 0; i < size; i++)
    {
        if (flights[i].duration < maxDuration) {
            printf("%s\t\t%s\t\t%s\t\t%.2f\n",
                   flights[i].flightNumber,
                   flights[i].departure,
```

```
                    flights[i].destination,
                    flights[i].duration);
        }
    }
}

void main()
{
    int numFlights;
    float maxDuration;
    printf("Enter the number of flights: ");
    scanf("%d", &numFlights);
    struct Flight flights[numFlights];
    for (int i = 0; i < numFlights; i++)
    {
        printf("\nEnter details for flight %d:\n", i + 1);
        printf("Flight Number: ");
        scanf("%s", flights[i].flightNumber);
        printf("Departure: ");
        scanf("%s", flights[i].departure);
        printf("Destination: ");
        scanf("%s", flights[i].destination);
        printf("Duration (in hours): ");
        scanf("%f", &flights[i].duration);
    }
    printf("\nEnter the maximum duration to filter flights (in hours): ");
    scanf("%f", &maxDuration);
    displayFlights(flights, numFlights, maxDuration);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($?) { gcc tempCodeRunnerFile
Enter the number of flights: 3

Enter details for flight 1:
Flight Number: 101
Departure: india
Destination: dubai
Duration (in hours): 5

Enter details for flight 2:
Flight Number: 102
Departure: dubai
Destination: england
Duration (in hours): 6

Enter details for flight 3:
Flight Number: 103
Departure: dubai
Destination: america
Duration (in hours): 7

Enter the maximum duration to filter flights (in hours): 6

Flights with duration less than 6.00 hours:
Flight No       Departure       Destination     Duration
101             india           dubai           5.00
PS D:\projects\quest\C>
```

```c
/*Define a structure to represent a term of a polynomial (coefficient and
exponent).
Write functions to add and multiply two polynomials.*/
#include <stdio.h>
#define MAX_TERMS 100
struct Term
{
    int coefficient;
    int exponent;
};

int addPolynomials(struct Term poly1[], int size1, struct Term poly2[],
int size2, struct Term result[]) {
    int i = 0, j = 0, k = 0;
    while (i < size1 && j < size2)
```

```
    {
        if (poly1[i].exponent > poly2[j].exponent)
        {
            result[k++] = poly1[i++];
        } else if (poly1[i].exponent < poly2[j].exponent)
        {
            result[k++] = poly2[j++];
        }
        else
        {
            result[k].coefficient = poly1[i].coefficient +
poly2[j].coefficient;
            result[k].exponent = poly1[i].exponent;
            k++;
            i++;
            j++;
        }
    }
    while (i < size1)
    {
        result[k++] = poly1[i++];
    }
    while (j < size2)
    {
        result[k++] = poly2[j++];
    }

    return k;
}
int multiplyPolynomials(struct Term poly1[], int size1, struct Term
poly2[], int size2, struct Term result[]) {
    int k = 0;
    for (int i = 0; i < size1; i++)
    {
        for (int j = 0; j < size2; j++)
        {
            int coeff = poly1[i].coefficient * poly2[j].coefficient;
            int exp = poly1[i].exponent + poly2[j].exponent;
            int found = 0;
            for (int l = 0; l < k; l++)
```

```
            {
                if (result[l].exponent == exp)
                {
                    result[l].coefficient += coeff;
                    found = 1;
                    break;
                }
            }
            if (!found)
            {
                result[k].coefficient = coeff;
                result[k].exponent = exp;
                k++;
            }
        }
    }
    return k;
}
void displayPolynomial(struct Term poly[], int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%d*x^%d", poly[i].coefficient, poly[i].exponent);
        if (i < size - 1)
        {
            printf(" + ");
        }
    }
    printf("\n");
}
void main() {
    struct Term poly1[] = {{3, 2}, {5, 1}, {6, 0}};
    struct Term poly2[] = {{1, 1}, {2, 0}};
    struct Term result[MAX_TERMS];
    int size1 = 3, size2 = 2, resultSize;
    printf("Polynomial 1: ");
    displayPolynomial(poly1, size1);
    printf("Polynomial 2: ");
    displayPolynomial(poly2, size2);
    resultSize = addPolynomials(poly1, size1, poly2, size2, result);
```

```
    printf("\nSum: ");
    displayPolynomial(result, resultSize);
    resultSize = multiplyPolynomials(poly1, size1, poly2, size2, result);
    printf("\nProduct: ");
    displayPolynomial(result, resultSize);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($
Polynomial 1: 3*x^2 + 5*x^1 + 6*x^0
Polynomial 2: 1*x^1 + 2*x^0

Sum: 3*x^2 + 6*x^1 + 8*x^0

Product: 3*x^3 + 11*x^2 + 16*x^1 + 12*x^0
PS D:\projects\quest\C>
```

```
/*Create a structure for a patient's medical record with fields for name,
age, diagnosis, and treatment.
Write a program to search for patients by diagnosis.*/
#include <stdio.h>
#include <string.h>
struct Patient {
    char name[50];
    int age;
    char diagnosis[100];
    char treatment[100];
};
void searchByDiagnosis(struct Patient patients[], int count, const char
*diagnosis)
{
    int found = 0;
    printf("Patients with diagnosis '%s':\n", diagnosis);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(patients[i].diagnosis, diagnosis) == 0)
        {
            found = 1;
            printf("\nName: %s\n", patients[i].name);
            printf("Age: %d\n", patients[i].age);
```

```
            printf("Diagnosis: %s\n", patients[i].diagnosis);
            printf("Treatment: %s\n", patients[i].treatment);
        }
    }
    if (!found)
    {
        printf("No patients found with the diagnosis '%s'.\n", diagnosis);
    }
}
void main() {
    int numPatients = 3;
    struct Patient patients[3] = {
        {"Alice Johnson", 45, "Diabetes", "Insulin Therapy"},
        {"Bob Smith", 34, "Hypertension", "Lifestyle Changes and
Medication"},
        {"Charlie Brown", 29, "Diabetes", "Dietary Management"}
    };
    char searchDiagnosis[100];
    printf("Enter the diagnosis to search for: ");
    fgets(searchDiagnosis, sizeof(searchDiagnosis), stdin);
    searchDiagnosis[strcspn(searchDiagnosis, "\n")] = '\0';
    searchByDiagnosis(patients, numPatients, searchDiagnosis);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if
Enter the diagnosis to search for: Diabetes
Patients with diagnosis 'Diabetes':

Name: Alice Johnson
Age: 45
Diagnosis: Diabetes
Treatment: Insulin Therapy

Name: Charlie Brown
Age: 29
Diagnosis: Diabetes
Treatment: Dietary Management
PS D:\projects\quest\C>
```

```c
/*Define a structure to store player information, including name, game
played, and score.
Write a program to display the top scorer for each game.
*/
#include <stdio.h>
#include <string.h>
struct Player
{
    char name[50];
    char game[50];
    int score;
};
void displayTopScorer(struct Player players[], int count)
{
    struct Player topScorers[100];
    int gameCount = 0;
    for (int i = 0; i < count; i++)
    {
        int found = 0;
        for (int j = 0; j < gameCount; j++)
```

```
        {
            if (strcmp(players[i].game, topScorers[j].game) == 0)
            {
                if (players[i].score > topScorers[j].score)
                {
                    topScorers[j] = players[i];
                }
                found = 1;
                break;
            }
        }
        if (!found)
        {
            topScorers[gameCount] = players[i];
            gameCount++;
        }
    }
    printf("\nTop scorer for each game:\n");
    for (int i = 0; i < gameCount; i++)
    {
        printf("Game: %s\n", topScorers[i].game);
        printf("Top Scorer: %s\n", topScorers[i].name);
        printf("Score: %d\n\n", topScorers[i].score);
    }
}
void main()
{
    int numPlayers = 5;
    struct Player players[] = {
        {"Alice", "Chess", 1200},
        {"Bob", "Football", 5},
        {"Charlie", "Chess", 1500},
        {"David", "Football", 10},
        {"Eve", "Chess", 1100}
    };
    displayTopScorer(players, numPlayers);
}
```

```
PS D:\projects\quest\c> cd "d:\projects\q

Top scorer for each game:
Game: Chess
Top Scorer: Charlie
Score: 1500

Game: Football
Top Scorer: David
Score: 10
```

```c
/*Create a structure to store information about a city, including name,
population, and area.
Write a program to calculate and display the population density of each
city.
*/
#include <stdio.h>
struct City
{
    char name[50];
    int population;
    float area;
};
void displayPopulationDensity(struct City cities[], int count)
{
    for (int i = 0; i < count; i++)
     {
        float density = cities[i].population / cities[i].area;
        printf("City: %s\n", cities[i].name);
        printf("Population: %d\n", cities[i].population);
        printf("Area: %.2f square kilometers\n", cities[i].area);
        printf("Population Density: %.2f people per square kilometer\n\n",
density);
    }
}
void main()
 {
    int numCities = 3;
    struct City cities[] = {
        {"New York", 8419600, 783.8},
```

```
        {"Los Angeles", 3980400, 1213.9},
        {"Chicago", 2716000, 606.1}
    };
    displayPopulationDensity(cities, numCities);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($?) { gcc temp
City: New York
Population: 8419600
Area: 783.80 square kilometers
Population Density: 10742.03 people per square kilometer

City: Los Angeles
Population: 3980400
Area: 1213.90 square kilometers
Population Density: 3279.02 people per square kilometer

City: Chicago
Population: 2716000
Area: 606.10 square kilometers
Population Density: 4481.11 people per square kilometer

PS D:\projects\quest\C>
```

```c
/*Define a structure for vehicle registration details, including
registration number, owner, make, and year.
Write a program to list all vehicles registered in a given year.*/
#include <stdio.h>
#include <string.h>
struct Vehicle
{
    char registrationNumber[15];
    char owner[50];
    char make[20];
    int year;
};
void listVehiclesByYear(struct Vehicle vehicles[], int count, int year)
{
    int found = 0;
    for (int i = 0; i < count; i++)
```

```
    {
        if (vehicles[i].year == year)
        {
            printf("Registration Number: %s\n",
vehicles[i].registrationNumber);
            printf("Owner: %s\n", vehicles[i].owner);
            printf("Make: %s\n", vehicles[i].make);
            printf("Year: %d\n\n", vehicles[i].year);
            found = 1;
        }
    }

    if (!found)
    printf("No vehicles found for the year %d.\n", year);
}

void main()
{
    struct Vehicle vehicles[] = {
        {"ABC123", "John Doe", "Toyota", 2020},
        {"XYZ456", "Alice Smith", "Honda", 2021},
        {"LMN789", "Bob Brown", "Ford", 2020},
        {"PQR234", "Charlie Davis", "Chevrolet", 2019}
    };
    int numVehicles = sizeof(vehicles) / sizeof(vehicles[0]);
    int searchYear;
    printf("Enter the year to search for registered vehicles: ");
    scanf("%d", &searchYear);
    listVehiclesByYear(vehicles, numVehicles, searchYear);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if (
Enter the year to search for registered vehicles: 2021
Registration Number: XYZ456
Owner: Alice Smith
Make: Honda
Year: 2021

PS D:\projects\quest\C>
```

```c
/*Create a structure to represent a menu item with fields for name, category, and price.
Write a program to display menu items in a specific category.*/
#include <stdio.h>
#include <string.h>
struct MenuItem {
    char name[50];
    char category[30];
    float price;
};
void displayMenuByCategory(struct MenuItem menu[], int count, const char* category)
{
    int found = 0;
    for (int i = 0; i < count; i++)
    {
        if (strcmp(menu[i].category, category) == 0)
        {
            printf("Name: %s\n", menu[i].name);
            printf("Price: $%.2f\n\n", menu[i].price);
            found = 1;
        }
    }

    if (!found)
    printf("No items found in the category '%s'.\n", category);
}
```

```c
int main()
{
    struct MenuItem menu[] = {
        {"Cheeseburger", "Main Course", 8.99},
        {"Pizza Margherita", "Main Course", 12.99},
        {"Caesar Salad", "Salad", 5.99},
        {"Grilled Chicken", "Main Course", 10.99},
        {"Fruit Salad", "Dessert", 4.99},
        {"Chocolate Cake", "Dessert", 6.99}
    };
    int numItems = sizeof(menu) / sizeof(menu[0]);
    char category[30];
    printf("Enter the category to search for (e.g., Main Course, Salad,
Dessert): ");
    fgets(category, sizeof(category), stdin);
    category[strcspn(category, "\n")] = '\0';
    displayMenuByCategory(menu, numItems, category);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($?) { gcc tempCodeRunnerFile.c -o tem
Enter the category to search for (e.g., Main Course, Salad, Dessert): Main Course
Name: Cheeseburger
Price: $8.99

Name: Pizza Margherita
Price: $12.99

Name: Grilled Chicken
Price: $10.99

PS D:\projects\quest\C>
```

```c
/*Define a structure for a sports team with fields for team name, sport,
number of players, and coach.
Write a program to display all teams playing a specific sport.*/
#include <stdio.h>
#include <string.h>
struct SportsTeam
{
    char teamName[50];
    char sport[30];
    int numPlayers;
```

```
    char coach[50];
};
void displayTeamsBySport(struct SportsTeam teams[], int count, const char*
sport)
 {
    int found = 0;
    for (int i = 0; i < count; i++)
    {
        if (strcmp(teams[i].sport, sport) == 0)
        {
            printf("Team Name: %s\n", teams[i].teamName);
            printf("Sport: %s\n", teams[i].sport);
            printf("Number of Players: %d\n", teams[i].numPlayers);
            printf("Coach: %s\n\n", teams[i].coach);
            found = 1;
        }
    }
    if (!found)
     printf("No teams found for the sport '%s'.\n", sport);
}

void main()
{
    struct SportsTeam teams[5] = {
        {"Team A", "Football", 11, "Coach John"},
        {"Team B", "Basketball", 5, "Coach Smith"},
        {"Team C", "Football", 11, "Coach Brown"},
        {"Team D", "Cricket", 11, "Coach Green"},
        {"Team E", "Basketball", 5, "Coach White"}
    };
    int numTeams = 5;
    char sport[30];
    printf("Enter the sport to search for (e.g., Football, Basketball,
Cricket): ");
    fgets(sport, sizeof(sport), stdin);
    sport[strcspn(sport, "\n")] = '\0';
    displayTeamsBySport(teams, numTeams, sport);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ; if ($?) { gcc tempCodeRunnerFile.
Enter the sport to search for (e.g., Football, Basketball, Cricket): Basketball
Team Name: Team B
Sport: Basketball
Number of Players: 5
Coach: Coach Smith

Team Name: Team E
Sport: Basketball
Number of Players: 5
Coach: Coach White

PS D:\projects\quest\C>
```

```c
/*Create a structure to store student marks in different subjects.
Write a program to calculate the total and percentage of marks for each
student.
*/
#include <stdio.h>
#include<string.h>
struct Student
{
    char name[50];
    int marks[5];
    int total;
    float percentage;
};

void calculateTotalAndPercentage(struct Student *s)
{
    s->total = 0;
    for (int i = 0; i < 5; i++)
        s->total += s->marks[i];
    s->percentage = (float)s->total / 5;
}
void main()
{
    struct Student students[3];
    for (int i = 0; i < 3; i++)
    {
        printf("Enter details for student %d:\n", i + 1);
```

```c
        printf("Name: ");
        fgets(students[i].name, sizeof(students[i].name), stdin);
        students[i].name[strcspn(students[i].name, "\n")] = '\0';
        printf("Enter marks for 5 subjects: ");
        for (int j = 0; j < 5; j++)
        {
            scanf("%d", &students[i].marks[j]);
        }
        getchar();
        calculateTotalAndPercentage(&students[i]);
        printf("\nStudent: %s\n", students[i].name);
        printf("Total Marks: %d\n", students[i].total);
        printf("Percentage: %.2f%%\n\n", students[i].percentage);
    }
}
```

```
Enter details for student 1:
Name: Rahul
Enter marks for 5 subjects: 50 34 55 13 76

Student: Rahul
Total Marks: 228
Percentage: 45.60%

Enter details for student 2:
Name: Gokul
Enter marks for 5 subjects: 67 89 65 77 98

Student: Gokul
Total Marks: 396
Percentage: 79.20%

Enter details for student 3:
Name: Kris
Enter marks for 5 subjects: 57 89 66 70 34

Student: Kris
Total Marks: 316
Percentage: 63.20%
```

```c
/*Define a structure for an e-commerce product with fields for product ID,
name, category, price, and stock.
Write a program to update the stock and calculate the total value of
products in stock.*/
#include <stdio.h>
struct Product
{
    int productID;
    char name[50];
    char category[30];
    float price;
    int stock;
};
void updateStock(struct Product *p, int quantity)
{
    p->stock += quantity;
    printf("Stock updated! New stock for product %s: %d\n", p->name,
p->stock);
}
float calculateTotalValue(struct Product p) {
    return p.price * p.stock;
}
void main()
{
    struct Product products[3] = {
        {101, "Laptop", "Electronics", 70000.0, 10},
        {102, "Smartphone", "Electronics", 25000.0, 15},
        {103, "Headphones", "Accessories", 3000.0, 50}
    };
    int option, productIndex, quantity;
    do {
        printf("\nE-Commerce Product Management System\n");
        printf("1. Update product stock\n");
        printf("2. Display total value of products in stock\n");
        printf("3. Exit\n");
        printf("Enter option: ");
        scanf("%d", &option);
        switch(option)
        {
```

```
            case 1:printf("Enter product index (0 for Laptop, 1 for
Smartphone, 2 for Headphones): ");
            scanf("%d", &productIndex);
            if (productIndex < 0 || productIndex > 2)
            {
                printf("Invalid product index!\n");
                break;
            }
            printf("Enter quantity to update: ");
            scanf("%d", &quantity);
            updateStock(&products[productIndex], quantity);
            break;

            case 2:printf("\nTotal value of products in stock:\n");
            for (int i = 0; i < 3; i++)
            printf("%s (ID: %d) - Total value: %.2f\n", products[i].name,
products[i].productID, calculateTotalValue(products[i]));
            break;

            case 3:printf("Exiting program...\n");
            break;

            default:printf("Invalid option! Please try again.\n");
            break;
        }
    } while (option!=3);
}
```

```
E-Commerce Product Management System
1. Update product stock
2. Display total value of products in stock
3. Exit
Enter option: 1
Enter product index (0 for Laptop, 1 for Smartphone, 2 for Headphones): 0
Enter quantity to update: 50
Stock updated! New stock for product Laptop: 60

E-Commerce Product Management System
1. Update product stock
2. Display total value of products in stock
3. Exit
Enter option: 2

Total value of products in stock:
Laptop (ID: 101) - Total value: 4200000.00
Smartphone (ID: 102) - Total value: 375000.00
Headphones (ID: 103) - Total value: 150000.00

E-Commerce Product Management System
1. Update product stock
2. Display total value of products in stock
3. Exit
Enter option: 3
Exiting program...
PS D:\projects\quest\C>
```

```c
/*Create a structure to store details of a music album, including album
name, artist, genre, and release year.
Write a program to display albums of a specific genre.*/
#include <stdio.h>
#include <string.h>
struct Album
{
    char albumName[50];
    char artist[50];
    char genre[30];
    int releaseYear;
};
```

```c
void displayAlbumsByGenre(struct Album albums[], int totalAlbums, const
char *genre)
{
    int found = 0;
    for (int i = 0; i < totalAlbums; i++)
    {
        if (strcmp(albums[i].genre, genre) == 0)
        {
            printf("Album Name: %s\n", albums[i].albumName);
            printf("Artist: %s\n", albums[i].artist);
            printf("Release Year: %d\n\n", albums[i].releaseYear);
            found = 1;
        }
    }
    if (!found)
    printf("No albums found in the genre: %s\n", genre);

}
void main()
{

    struct Album albums[5] = {
        {"Album1", "Artist1", "Rock", 2001},
        {"Album2", "Artist2", "Pop", 2005},
        {"Album3", "Artist3", "Rock", 2008},
        {"Album4", "Artist4", "Jazz", 2010},
        {"Album5", "Artist5", "Pop", 2015}
    };
    char genre[30];
    printf("Enter genre to display albums: ");
    scanf("%s", genre);
    printf("\nAlbums of genre: %s\n", genre);
    displayAlbumsByGenre(albums, 5, genre);
}
```

```
Enter genre to display albums: Pop

Albums of genre: Pop
Album Name: Album2
Artist: Artist2
Release Year: 2005

Album Name: Album5
Artist: Artist5
Release Year: 2015

PS D:\projects\quest\C>
```

```c
/*Define a structure for a cinema ticket with fields for movie name, seat
number, and price.
Write a program to book tickets and display the total revenue generated.*/
#include <stdio.h>
#include <string.h>
struct CinemaTicket {
    char movieName[50];
    int seatNumber;
    float price;
};
void bookTicket(struct CinemaTicket *ticket, const char *movie, int seat,
float price)
{
    strcpy(ticket->movieName, movie);
    ticket->seatNumber = seat;
    ticket->price = price;
}

void displayTicket(struct CinemaTicket ticket)
{
    printf("Movie: %s\n", ticket.movieName);
    printf("Seat Number: %d\n", ticket.seatNumber);
    printf("Ticket Price: %.2f\n", ticket.price);
}
float calculateTotalRevenue(struct CinemaTicket tickets[], int count)
```

```
{
    float totalRevenue = 0;
    for (int i = 0; i < count; i++)
    totalRevenue += tickets[i].price;

    return totalRevenue;
}
void main()
{
    int ticketCount = 0;
    struct CinemaTicket tickets[100];
    int option, seat;
    char movie[50];
    float price;
    do {
        printf("Cinema Ticket Booking System\n");
        printf("1. Book Ticket\n");
        printf("2. Display All Booked Tickets\n");
        printf("3. Display Total Revenue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &option);

        switch(option) {
            case 1:printf("Enter movie name: ");
                getchar();
                fgets(movie, sizeof(movie), stdin);
                movie[strcspn(movie, "\n")] = 0;
                printf("Enter seat number: ");
                scanf("%d", &seat);
                printf("Enter ticket price: ");
                scanf("%f", &price);
                bookTicket(&tickets[ticketCount], movie, seat, price);
                ticketCount++;
                printf("Ticket booked successfully!\n");
                break;
            case 2: if (ticketCount == 0)
                printf("No tickets booked yet!\n");
                else
                {
```

```
                    printf("\nBooked Tickets:\n");
                    for (int i = 0; i < ticketCount; i++)
                    {
                        displayTicket(tickets[i]);
                        printf("\n");
                    }
                }
                break;
            case 3:if(ticketCount == 0)
                printf("No tickets booked yet!\n");
             else
             {
                float totalRevenue = calculateTotalRevenue(tickets,
ticketCount);
                printf("Total Revenue: %.2f\n", totalRevenue);
             }
            break;
            case 4:printf("Exiting the system...\n");
            break;
            default:printf("Invalid choice. Please try again.\n");
            break;
        }
    } while (option != 4);
}
```

```
2. Display All Booked Tickets
3. Display Total Revenue
4. Exit
Enter your choice: 1
Enter movie name: Hero
Enter seat number: 2
Enter ticket price: 200
Ticket booked successfully!
Cinema Ticket Booking System
1. Book Ticket
2. Display All Booked Tickets
3. Display Total Revenue
4. Exit
Enter your choice: 1
Enter movie name: Rio
Enter seat number: 3
Enter ticket price: 200
Ticket booked successfully!
Cinema Ticket Booking System
1. Book Ticket
2. Display All Booked Tickets
3. Display Total Revenue
4. Exit
Enter your choice: 2

Booked Tickets:
Movie: Hero
Seat Number: 2
Ticket Price: 200.00

Movie: Rio
Seat Number: 3
Ticket Price: 200.00

Cinema Ticket Booking System
1. Book Ticket
2. Display All Booked Tickets
3. Display Total Revenue
4. Exit
Enter your choice: 3
Total Revenue: 400.00
```