Assignment 13

1. Student (rollNo, name, marks)

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

typedef struct Student
{
    int rollNo;
    char name[25];
    int marks;
} Student;

void storeStudent(Student *std, int *idx)
{
    printf("\nEnter Roll No:");
    scanf("%d", &std[*idx].rollNo);
    printf("\nEnter Name:");
    fflush(stdin);
    gets(std[*idx].name);
    printf("\nEnter Marks:");
    scanf("%d", &std[*idx].marks);
}

void displayStudent(Student *std, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nRoll No: %d", std[i].rollNo);
        printf("\nName: %s", std[i].name);
        printf("\nMarks: %d", std[i].marks);
    }
}

void displayOne(Student *std)
{
    printf("\nRoll No: %d", std->rollNo);
    printf("\nName: %s", std->name);
    printf("\nMarks: %d", std->marks);
}

int seerachIndex(Student *s, int rn, int *idx)
{
    for (int i = 0; i < *idx; i++)
    {
        if (s[i].rollNo == rn)
        {
            return i;
        }
```

```
    }
    return -1;
}

void main()
{
    Student s[10];
    int idx = 0;
    printf("\nHow many students do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeStudent(s, &idx);
    }
    displayStudent(s, &idx);
    printf("\nEnter the Roll no of student you want to search");
    int rn;
    scanf("%d", &rn);
    rn = seerachIndex(s, rn, &idx);
    if (rn == -1)
    {
        printf("\nNo stud Found");
    }
    else
    {
        displayOne(&s[rn]);
    }
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\..... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
How many students do you want to Store : 5
Enter Roll No:1
Enter Name:Bhagvat
Enter Marks:89
Enter Roll No:2
Enter Name:p
Enter Marks:78
Enter Roll No:3
Enter Name:jay
Enter Marks:98
Enter Roll No:4
Enter Name:pk
Enter Marks:45
Enter Roll No:5
Enter Name:raju
Enter Marks:67

 Entered data

Roll No: 1
Name: Bhagvat
Marks: 89

Roll No: 2
```

Name: p
Marks: 78

Roll No: 3
Name: jay
Marks: 98

Roll No: 4
Name: pk
Marks: 45

Roll No: 5
Name: raju
Marks: 67
Enter the Roll no of student you want to search1

Roll No: 1
Name: Bhagvat
Marks: 89
PS C:\Code>

```c
#include <stdio.h>
#include <string.h>
typedef struct Student
{
    int rollNo;
    char name[25];
    int marks;
} Student;

Student storeStudentPV()
{
    Student tmp;
    printf("Enter Student Roll no:");
    scanf("%d", &tmp.rollNo);
    printf("Enter Student Name:");
    scanf("%s", tmp.name);
    printf("Enter Student Marks:");
    scanf("%d", &tmp.marks);
    return tmp;
}

void displayStudentPV(Student s)
{
    printf("\nRoll no of student : %d", s.rollNo);
    printf("\nNameof student : %s", s.name);
    printf("\nMarks of student : %d", s.marks);
}

void main()
{
    Student s1;
    s1 = storeStudentPV();
    displayStudentPV(s1);
```

```
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\....\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter Student Roll no:1
Enter Student Name:Bhagvat
Enter Student Marks:98

Roll no of student : 1
Nameof student : Bhagvat
Marks of student : 98
PS C:\Code>
```

2. Employee (id, name, salary)

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

typedef struct Employee
{
    int empId;
    char empName[25];
    int empSalary;
} Employee;

void storeEmployee(Employee *emp, int *idx)
{
    printf("\nEnter EmpID:");
    scanf("%d", &emp[*idx].empId);
    printf("\nEnter empempName:");
    fflush(stdin);
    gets(emp[*idx].empName);
    printf("\nEnter empSalary:");
    scanf("%d", &emp[*idx].empSalary);
}

void displayEmployee(Employee *emp, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nEMP ID: %d", emp[i].empId);
        printf("\nempName: %s", emp[i].empName);
        printf("\nempSalary: %d", emp[i].empSalary);
    }
}

Employee storeEmployeePV()
{
    Employee tmp;
    printf("Enter Employee EMP ID:");
    scanf("%d", &tmp.empId);
```

```
    printf("Enter Employee empName:");
    fflush(stdin);
    gets(tmp.empName);
    printf("Enter Employee empSalary:");
    scanf("%d", &tmp.empSalary);
    return tmp;
}

void displayEmployeePV(Employee e)
{
    printf("\nEMP ID of Employee : %d", e.empId);
    printf("\nempNameof Employee : %s", e.empName);
    printf("\nempSalary of Employee : %d", e.empSalary);
}

int searchEmpById(Employee *e, int id, int *idx)
{
    for (int i = 0; i < *idx; i++)
    {
        if (e[i].empId == id)
        {
            return i;
        }
    }
    return -1;
}

void printByAddress(Employee *e)
{
    printf("\n");
    printf("\nEMP ID: %d", e->empId);
    printf("\nempName: %s", e->empName);
    printf("\nempSalary: %d", e->empSalary);
}

void main()
{
    Employee e1;
    e1 = storeEmployeePV();
    displayEmployeePV(e1);

    Employee s[20];
    int idx = 0;
    printf("\nHow many Employees do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeEmployee(s, &idx);
    }
    displayEmployee(s, &idx);

    printf("\nEnter Employee  ID to search : ");
```

```
    int id;
    scanf("%d", &id);
    int res = searchEmpById(s, id, &idx);
    if (res == -1)
    {
        printf("\n Not found");
    }
    else
    {
        printByAddress(&s[res]);
    }
}
```

Output:
PS C:\Code> & 'c:\Users\bhagv\.vscode\.....\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter Employee EMP ID:123
Enter Employee empName:Bhagvat Mutthe
Enter Employee empSalary:500000

EMP ID of Employee : 123
empNameof Employee : Bhagvat Mutthe
empSalary of Employee : 500000
How many Employees do you want to Store : 3

Enter EmpID:11
Enter empempName:Pravin
Enter empSalary:50000
Enter EmpID:43
Enter empempName:ap
Enter empSalary:25656
Enter EmpID:67
Enter empempName:pk
Enter empSalary:3448

 Entered data

EMP ID: 11
empName: Pravin
empSalary: 50000

EMP ID: 43
empName: ap
empSalary: 25656

EMP ID: 67
empName: pk
empSalary: 3448
Enter Employee ID to search : 123

 Not found
PS C:\Code>

3. Admin (id, name, salary, allowance)

```c
#include <stdio.h>
#include <string.h>
int size = 10;
typedef struct Admin
{
    int aId;
    char name[25];
    int aSalary;
    float allowence;
} Admin;

void storeAdmin(Admin *adm, int *idx)
{
    printf("\nEnter Admin ID:");
    scanf("%d", &adm[*idx].aId);
    printf("\nEnter Name:");
    fflush(stdin);
    gets(adm[*idx].name);
    printf("\nEnter aSalary:");
    scanf("%d", &adm[*idx].aSalary);
    printf("\nEnter Allowence:");
    scanf("%f", &adm[*idx].allowence);
}

void displayAdmin(Admin *adm, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nAdmin ID: %d", adm[i].aId);
        printf("\nName: %s", adm[i].name);
        printf("\naSalary: %d", adm[i].aSalary);
        printf("\nAllowence: %.2f", adm[i].allowence);
    }
}

Admin storeByValue()
{
    Admin temp;
    printf("\nEnter Admin ID:");
    scanf("%d", &temp.aId);
    printf("\nEnter Name:");
    fflush(stdin);
    gets(temp.name);
    printf("\nEnter aSalary:");
    scanf("%d", &temp.aSalary);
    printf("\nEnter Allowence:");
    scanf("%f", &temp.allowence);
    return temp;
}

void displayByValue(Admin a)
```

```
{
    printf("\n");
    printf("\nAdmin ID: %d", a.aId);
    printf("\nName: %s", a.name);
    printf("\naSalary: %d", a.aSalary);
    printf("\nAllowence: %.2f", a.allowence);
}

int searchByid(Admin *a, int id)
{
    for (int i = 0; i < size; i++)
    {
        if (a[i].aId == id)
        {
            return i;
        }
    }
    return -1;
}

void printByAddr(Admin *a)
{
    printf("\n");
    printf("\nAdmin ID: %d", a->aId);
    printf("\nName: %s", a->name);
    printf("\naSalary: %d", a->aSalary);
    printf("\nAllowence: %.2f", a->allowence);
}

void main()
{
    Admin a1;
    a1 = storeByValue();
    displayByValue(a1);

    Admin s[size];
    int idx = 0;
    printf("\nHow many Admins do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeAdmin(s, &idx);
    }
    displayAdmin(s, &idx);

    printf("\nEnter id of admin you want to search: ");
    int id, ind;
    scanf("%d", &id);
    ind = searchByid(s, id);

    if (ind == -1)
    {
```

```
        printf("\nNot Found");
    }
    else
    {
        printByAddr(&s[ind]);
    }
}
```

Output:
C:\Code> cmd /C "c:\Users\bhagv\.vscode\.....\TDM-GCC-64\bin\gdb.exe --interpreter=mi "
Enter Admin ID:12
Enter Name:afda
Enter aSalary:352443
Enter Allowence:323

Admin ID: 12
Name: afda
aSalary: 352443
Allowence: 323.00
How many Admins do you want to Store : 3
Enter Admin ID:752
Enter Name:kjbhlkjafn
Enter aSalary:1543324
Enter Allowence:1346
Enter Admin ID:123
Enter Name:jlhnkjjn
Enter aSalary:12545
Enter Allowence:325
Enter Admin ID:32
Enter Name:l;knkn
Enter aSalary:23546
Enter Allowence:234

Entered data

Admin ID: 752
Name: kjbhlkjafn
aSalary: 1543324
Allowence: 1346.00

Admin ID: 123
Name: jlhnkjjn
aSalary: 12545
Allowence: 325.00

Admin ID: 32
Name: l;knkn
aSalary: 23546
Allowence: 234.00
Enter id of admin you want to search: 123

Admin ID: 123
Name: jlhnkjjn
aSalary: 12545

Allowence: 325.00
C:\Code>

4. HR (id, name, salary, commission)

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

typedef struct HR
{
    int hrID;
    char name[25];
    int Salary;
    int commission;
} HR;

void storeHR(HR *hrS, int *idx)
{
    printf("\nEnter HR Id:");
    scanf("%d", &hrS[*idx].hrID);
    printf("\nEnter Name:");
    fflush(stdin);
    gets(hrS[*idx].name);
    printf("\nEnter Salary:");
    scanf("%d", &hrS[*idx].Salary);
    printf("\nEnter Commission:");
    scanf("%d", &hrS[*idx].commission);
}

void displayHR(HR *hrS, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nHR Id: %d", hrS[i].hrID);
        printf("\nName: %s", hrS[i].name);
        printf("\nSalary: %d", hrS[i].Salary);
        printf("\nCommission: %d", hrS[i].commission);
    }
}

HR storeByValue()
{
    HR temp;
    printf("\nEnter HR Id:");
    scanf("%d", &temp.hrID);
    printf("\nEnter Name:");
    fflush(stdin);
    gets(temp.name);
    printf("\nEnter Salary:");
    scanf("%d", &temp.Salary);
```

```
    printf("\nEnter Commission:");
    scanf("%d", &temp.commission);
    return temp;
}

void diaplayByValue(HR h)
{
    printf("\n");
    printf("\nHR Id: %d", h.hrID);
    printf("\nName: %s", h.name);
    printf("\nSalary: %d", h.Salary);
    printf("\nCommission: %d", h.commission);
}

int searcById(HR *h, int id, int size)
{
    for (int i = 0; i < size; i++)
    {
        if (h[i].hrID == id)
        {
            return i;
        }
    }
    return -1;
}

void displayByAddr(HR *h)
{
    printf("\n");
    printf("\nHR Id: %d", h->hrID);
    printf("\nName: %s", h->name);
    printf("\nSalary: %d", h->Salary);
    printf("\nCommission: %d", h->commission);
}
void main()
{
    HR h;
    h = storeByValue();
    diaplayByValue(h);
    HR s[10];
    int idx = 0;
    printf("\nHow many HRs do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeHR(s, &idx);
    }
    displayHR(s, &idx);

    int id, ind;
    printf("\nEnter the id of HR you want to search : ");
    scanf("%d", &id);
```

```
    ind = searcById(s, id, n);

    if (ind == -1)
    {
        printf("\nNot Found");
    }
    else
    {
        displayByAddr(&s[ind]);
    }
}
```

Output:
C:\Code> cmd /C "c:\Users\bhagv\.vscode\...\TDM-GCC-64\bin\gdb.exe --interpreter=mi "

Enter HR Id:1
Enter Name:shytf
Enter Salary:15213
Enter Commission:23121

HR Id: 1
Name: shytf
Salary: 15213
Commission: 23121
How many HRs do you want to Store : 3

Enter HR Id:11
Enter Name:xfghsd
Enter Salary:321
Enter Commission:2132
Enter HR Id:12
Enter Name:dfgstgsg
Enter Salary:32145
Enter Commission:321
Enter HR Id:13
Enter Name:sghs
Enter Salary:32156
Enter Commission:5646

Entered data

HR Id: 11
Name: xfghsd
Salary: 321
Commission: 2132

HR Id: 12
Name: dfgstgsg
Salary: 32145
Commission: 321

HR Id: 13
Name: sghs
Salary: 32156
Commission: 5646
Enter the id of HR you want to search : 11

HR Id: 11
Name: xfghsd
Salary: 321
Commission: 2132
C:\Code>

5. SalesManager (id, name, salary, incentive, target)

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

// SalesManager structure with ID, name, salary, incentive, and target
typedef struct SalesManager
{
    int smID;
    char name[25];
    int salary;
    int incentive;
    int target;
} SalesManager;

// Function to store SalesManager details by reference
void storeSalesManager(SalesManager *saleMgr, int *idx)
{
    printf("\nEnter SM ID: ");
    scanf("%d", &saleMgr[*idx].smID);
    printf("\nEnter Name: ");
    fflush(stdin);
    gets(saleMgr[*idx].name);
    printf("\nEnter salary: ");
    scanf("%d", &saleMgr[*idx].salary);
    printf("\nEnter Incentive: ");
    scanf("%d", &saleMgr[*idx].incentive);
    printf("\nEnter Target: ");
    scanf("%d", &saleMgr[*idx].target);
}

// Function to display SalesManager details by reference
void displaySalesManager(SalesManager *saleMgr, int *idx)
{
    printf("\nEntered data:");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nSM ID: %d", saleMgr[i].smID);
        printf("\nName: %s", saleMgr[i].name);
        printf("\nSalary: %d", saleMgr[i].salary);
        printf("\nIncentive: %d", saleMgr[i].incentive);
        printf("\nTarget: %d", saleMgr[i].target);
    }
}
```

```c
// Function to store SalesManager details by value
SalesManager storeByVal()
{
    SalesManager temp;
    printf("\nEnter SM ID: ");
    scanf("%d", &temp.smID);
    printf("\nEnter Name: ");
    fflush(stdin);
    gets(temp.name);
    printf("\nEnter salary: ");
    scanf("%d", &temp.salary);
    printf("\nEnter Incentive: ");
    scanf("%d", &temp.incentive);
    printf("\nEnter Target: ");
    scanf("%d", &temp.target);
    return temp;
}

// Function to display SalesManager details by value
void displayByVal(SalesManager s)
{
    printf("\n");
    printf("\nSM ID: %d", s.smID);
    printf("\nName: %s", s.name);
    printf("\nSalary: %d", s.salary);
    printf("\nIncentive: %d", s.incentive);
    printf("\nTarget: %d", s.target);
}

// Function to search SalesManager by ID
int searcById(SalesManager *s, int id, int size)
{
    for (int i = 0; i < size; i++)
    {
        if (s[i].smID == id)
        {
            return i;
        }
    }
    return -1;
}

// Function to display SalesManager details by address
void displayByAddr(SalesManager *s)
{
    printf("\n");
    printf("\nSM ID: %d", s->smID);
    printf("\nName: %s", s->name);
    printf("\nSalary: %d", s->salary);
    printf("\nIncentive: %d", s->incentive);
    printf("\nTarget: %d", s->target);
}
```

```c
// Main function
void main()
{
    SalesManager sm;
    // Store and display details using value-based functions
    sm = storeByVal();
    displayByVal(sm);

    SalesManager s[10];
    int idx = 0;
    printf("\nHow many SalesManagers do you want to store: ");
    int n;
    scanf("%d", &n);

    // Loop to store multiple SalesManager details
    for (int i = 0; i < n; i++, idx++)
    {
        storeSalesManager(s, &idx);
    }

    // Display details of all SalesManagers
    displaySalesManager(s, &idx);

    // Searching for a SalesManager by ID
    int id, ind;
    printf("\nEnter the SM ID you want to search: ");
    scanf("%d", &id);
    ind = searcById(s, id, n);

    // Check if SalesManager was found
    if (ind == -1)
    {
        printf("\nNot Found");
    }
    else
    {
        displayByAddr(&s[ind]);
    }
}
```

Output: C:\Code> cmd /C "c:\Users\bhagv\.vscode\.....\ TDM-GCC-64\bin\gdb.exe --interpreter=mi "

Enter SM ID: 1
Enter Name: afg
Enter salary: 3254
Enter Incentive: 54
Enter Target: 56

SM ID: 1
Name: afg
Salary: 3254
Incentive: 54

Target: 56
How many SalesManagers do you want to store: 3

Enter SM ID: 12
Enter Name: asdga
Enter salary: 6154
Enter Incentive: 15646
Enter Target: 65
Enter SM ID: 21
Enter Name: fdgsdg
Enter salary: 5463
Enter Incentive: 2156
Enter Target: 654
Enter SM ID: 564651

Enter Name: fgsdg
Enter salary: 5456
Enter Incentive: 65461
Enter Target: 51

Entered data:
SM ID: 12
Name: asdga
Salary: 6154
Incentive: 15646
Target: 65

SM ID: 21
Name: fdgsdg
Salary: 5463
Incentive: 2156
Target: 654

SM ID: 564651
Name: fgsdg
Salary: 5456
Incentive: 65461
Target: 51
Enter the SM ID you want to search: 21

SM ID: 21
Name: fdgsdg
Salary: 5463
Incentive: 2156
Target: 654
C:\Code>

6. Date (date, month, year)

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

// Date structure with Day, Month, and Year
typedef struct Date
{
```

```
    int DD;
    int MM;
    int YYYY;
} Date;

// Function to store Date by reference
void storeDate(Date *dte, int *idx)
{
    printf("\nEnter DD: ");
    scanf("%d", &dte[*idx].DD);
    printf("\nEnter MM: ");
    scanf("%d", &dte[*idx].MM);
    printf("\nEnter YYYY: ");
    scanf("%d", &dte[*idx].YYYY);
}

// Function to display all stored Dates by reference
void displayDate(Date *dte, int *idx)
{
    printf("\nEntered data:");
    for (int i = 0; i < *idx; i++)
    {
        printf("\nDate: %d/%d/%d", dte[i].DD, dte[i].MM, dte[i].YYYY);
    }
}

// Function to search Date by Year (YYYY)
int searchByYYYY(Date *d, int yyyy, int size)
{
    for (int i = 0; i < size; i++)
    {
        if (d[i].YYYY == yyyy)
        {
            return i;
        }
    }
    return -1;
}

// Function to display Date by address
void displayByAddr(Date *d)
{
    printf("\n");
    printf("\nDate: %d/%d/%d", d->DD, d->MM, d->YYYY);
}

Date storeByValue()
{
    Date temp;
    return temp;
}

void displayByValue(Date d)
```

```
{
    printf("\nDate: %d/%d/%d", d.DD, d.MM, d.YYYY);
}

// Main function
void main()
{
    Date dob;
    dob = storeByValue();
    displayByValue(dob);

    Date s[10];
    int idx = 0;

    // Input number of dates to store
    printf("\nHow many Dates do you want to store: ");
    int n;
    scanf("%d", &n);

    // Loop to store multiple Dates
    for (int i = 0; i < n; i++, idx++)
    {
        storeDate(s, &idx);
    }

    // Display all stored Dates
    displayDate(s, &idx);

    int yyyy, ind;

    // Searching for a Date by only Year (YYYY)
    printf("\nEnter the Year (YYYY) you want to search: ");
    scanf("%d", &yyyy);

    ind = searchByYYYY(s, yyyy, n);

    // Check if a Date with the given year was found
    if (ind == -1)
    {
        printf("\nDate with the specified year not found.");
    }
    else
    {
        displayByAddr(&s[ind]);
    }
}
```

Output:

C:\Code> cmd /C "c:\Users\bhagv\.vscode\.....\TDM-GCC-64\bin\gdb.exe --interpreter=mi "

Enter DD: 12
Enter MM: 2

```
Enter YYYY: 2024
Date: 12/2/2024
How many Dates do you want to store: 5
Enter DD: 11
Enter MM: 10
Enter YYYY: 2002
Enter DD: 24
Enter MM: 11
Enter YYYY: 2002
Enter DD: 16
Enter MM: 09
Enter YYYY: 2001
Enter DD: 09
Enter MM: 05
Enter YYYY: 2000
Enter DD: 18
Enter MM: 10
Enter YYYY: 1970

Entered data:
Date: 11/10/2002
Date: 24/11/2002
Date: 16/9/2001
Date: 9/5/2000
Date: 18/10/1970
Enter the Year (YYYY) you want to search: 2000
Date: 9/5/2000
C:\Code>
```

7. Time (hour, min, sec)

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

typedef struct Time
{
    int HH;
    int MM;
    int SS;
} Time;

void storeTime(Time *dte, int *idx)
{
    printf("\nEnter HH:");
    scanf("%d", &dte[*idx].HH);
    printf("\nEnter MM:");
    scanf("%d", &dte[*idx].MM);
    printf("\nEnter SS:");
    scanf("%d", &dte[*idx].SS);
}

void displayTime(Time *dte, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
```

```
    {
        printf("\n");
        printf("\nTime : %d:%d:%d", dte[i].HH, dte[i].MM, dte[i].SS);
    }
}

Time storeTimeByVal()
{
    Time temp;
    printf("\nEnter HH:");
    scanf("%d", &temp.HH);
    printf("\nEnter MM:");
    scanf("%d", &temp.MM);
    printf("\nEnter SS:");
    scanf("%d", &temp.SS);

    return temp;
}
void printTimeByVal(Time t)
{
    printf("\nTime(hh:mm:ss) = %d:%d:%d", t.HH, t.MM, t.SS);
}

int searchTimeByAddr(Time *t, int size, int HH)
{
    for (int i = 0; i < size; i++)
    {
        if (t[i].HH == HH)
        {
            return i;
        }
    }
    return -1;
}

void printByAddress(Time *t)
{
    printf("\nTime(hh:mm:ss) = %d:%d:%d", t->HH, t->MM, t->SS);
}

void main()
{
    Time t1;
    t1 = storeTimeByVal();
    printTimeByVal(t1);
    Time s[10];
    int idx = 0;
    printf("\nHow many Times do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeTime(s, &idx);
```

```
    }
    displayTime(s, &idx);
    printf("\nEnter Hour you want to serach : ");
    int hr;
    scanf("%d", &hr);
    int x = searchTimeByAddr(s, n, hr);
    printByAddress(&s[x]);
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\.....\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter HH:10
Enter MM:12
Enter SS:3
Time(hh:mm:ss) = 10:12:3
How many Times do you want to Store : 3
Enter HH:21
Enter MM:34
Enter SS:54
Enter HH:57
Enter MM:32
Enter SS:65
Enter HH:23
Enter MM:453
Enter SS:23

 Entered data

Time : 21:34:54
Time : 57:32:65
Time : 23:453:23
Enter Hour you want to serach : 21
Time(hh:mm:ss) = 21:34:54
PS C:\Code>
```

8. Distance (feet, inch)

```
#include <stdio.h>
#include <string.h>
// Distance ( feet, inch)
typedef struct Distance
{
    int feet;
    int inch;
} Distance;

void storeDistance(Distance *dist, int *idx)
{
    printf("\nEnter Feet:");
    scanf("%d", &dist[*idx].feet);
    printf("\nEnter inch:");
    scanf("%d", &dist[*idx].inch);
}

void displayDistance(Distance *dist, int *idx)
```

```
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nFeet: %d", dist[i].feet);
        printf("\ninch: %d", dist[i].inch);
    }
}

Distance storeByVal()
{
    Distance dist;
    printf("\nEnter Feet:");
    scanf("%d", &dist.feet);
    printf("\nEnter inch:");
    scanf("%d", &dist.inch);
    return dist;
}

void printByVal(Distance d)
{
    printf("\ninch: %d \nFeet: %d", d.inch, d.feet);
}

int searchByAddress(Distance *d, int size, int val)
{
    for (int i = 0; i < size; i++)
    {
        if (d[i].feet == val)
        {
            return i;
        }
    }
    return -1;
}

void printByAdd(Distance *d)
{
    printf("\nInch: %d \nFeet: %d", d->inch, d->feet);
}

void main()
{
    Distance d;
    d = storeByVal();
    printByVal(d);
    Distance s[10];
    int idx = 0;
    printf("\nHow many Distances do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
```

```
    {
        storeDistance(s, &idx);
    }
    displayDistance(s, &idx);
    printf("\nEnter the Feet to search ");
    int f;
    scanf("%d", &f);
    int x = searchByAddress(s, n, f);
    printByAdd(&s[x]);
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\...\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter Feet:12
Enter inch:23
inch: 23
Feet: 12
How many Distances do you want to Store : 3
Enter Feet:12
Enter inch:34
Enter Feet:56
Enter inch:78
Enter Feet:548
Enter inch:90

 Entered data

Feet: 12
inch: 34

Feet: 56
inch: 78

Feet: 548
inch: 90
Enter the Feet to search 12

Inch: 34
Feet: 12
PS C:\Code>
```

9. Complex (real, imaginary)

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Complex
{
    double real;
    double imaginary;
} Complex;

Complex getNumsByVal()
{
    Complex c;
    printf("Enter Real Part :");
    scanf("%d", &c.real);
```

```
    printf("Enter Imaginory Part :");
    scanf("%d", &c.imaginary);
    return c;
}

void showNumsByVal(Complex c)
{
    printf("\nYour imaginary no is : %d+%di", c.real, c.imaginary);
}

void storeArrComplex(Complex *c, int s)
{
    for (int i = 0; i < s; i++)
    {
        printf("\nEnter %d complex no.", i + 1);
        printf("\nEnter Imaginary part");
        scanf("%d", &c[i].imaginary);
        printf("\nEnter Real part");
        scanf("%d", &c[i].real);
    }
}

void displayArrComplex(Complex *c, int s)
{
    for (int i = 0; i < s; i++)
    {
        printf("\nYour imaginary %d  no is : %d+%di", i + 1, c[i].real, c[i].imaginary);
    }
}

int searchComplex(Complex *c, int s, int r)
{
    for (int i = 0; i < s; i++)
    {
        if (c[i].real == r)
        {
            return i;
        }
    }
    return -1;
}

void printIndexedElement(Complex *c)
{
    printf("\nYour imaginary no is : %d+%di", c->real, c->imaginary);
}
void main()
{
    Complex comp;
    comp = getNumsByVal();
    showNumsByVal(comp);

    int size, r;
```

```
    printf("\nEnter How many complex nums you want to store");
    scanf("%d", &size);
    Complex cArr[size];
    storeArrComplex(cArr, size);
    displayArrComplex(cArr, size);

    printf("\nEnter the num to search in array : (enter real part)");
    scanf("%d", &r);
    int idx = searchComplex(cArr, size, r);
    printf("\n%d is the index of the number you searched.", idx + 1);

    printIndexedElement(&cArr[idx]);
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\....\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter Real Part :123
Enter Imaginory Part :34

Your imaginary no is : 123+34i
Enter How many complex nums you want to store4
Enter 1 complex no.
Enter Imaginary part12
Enter Real part34
Enter 2 complex no.
Enter Imaginary part67
Enter Real part34
Enter 3 complex no.
Enter Imaginary part98
Enter Real part321
Enter 4 complex no.
Enter Imaginary part967
Enter Real part214

Your imaginary 1  no is : 34+12i
Your imaginary 2  no is : 34+67i
Your imaginary 3  no is : 321+98i
Your imaginary 4  no is : 214+967i
Enter the num to search in array : (enter real part)34
1 is the index of the number you searched.
Your imaginary no is : 34+12i
PS C:\Code>
```

10. Product (id, name, quantity, price)

```
#include <stdio.h>
#include <string.h>
// Product ( ID, quantiy)
typedef struct Product
{
    int ID;
    int quantiy;
    int price;
```

```
    char name[20];
} Product;

void storeProduct(Product *prod, int *idx)
{
    printf("\nEnter ID:");
    scanf("%d", &prod[*idx].ID);
    printf("\nEnter Name of product:");
    fflush(stdin);
    gets(prod[*idx].name);
    printf("\nEnter quantiy:");
    scanf("%d", &prod[*idx].quantiy);
    printf("\nEnter Price:");
    scanf("%d", &prod[*idx].price);
}

void displayProduct(Product *prod, int *idx)
{
    printf("\n Entered data");
    for (int i = 0; i < *idx; i++)
    {
        printf("\n");
        printf("\nID: %d", prod[i].ID);
        printf("\nName: %s", prod[i].name);
        printf("\nQuantiy: %d", prod[i].quantiy);
        printf("\nPrice: %d", prod[i].price);
    }
}

Product storeByVal()
{
    Product temp;

    printf("\nEnter ID:");
    scanf("%d", &temp.ID);
    printf("\nEnter Name of product:");
    fflush(stdin);
    gets(temp.name);
    printf("\nEnter quantiy:");
    scanf("%d", &temp.quantiy);
    printf("\nEnter Price:");
    scanf("%d", &temp.price);

    return temp;
}

void printByVal(Product p)
{
    printf("\nID: %d", p.ID);
    printf("\nName: %s", p.name);
    printf("\nQuantiy: %d", p.quantiy);
    printf("\nPrice: %d", p.price);
}
```

```
int searchByAddress(Product *p, int key, int *max)
{
    for (int i = 0; i < *max; i++)
    {
        if (p[i].ID == key)
        {
            return i;
        }
    }
    return -1;
}

void printByAddr(Product *p)
{
    printf("\nID: %d", p->ID);
    printf("\nName: %s", p->name);
    printf("\nQuantiy: %d", p->quantiy);
    printf("\nPrice: %d", p->price);
}

void main()
{
    Product p1;
    p1 = storeByVal();
    printByVal(p1);

    Product s[10];
    int idx = 0;
    printf("\nHow many Products do you want to Store : ");
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++, idx++)
    {
        storeProduct(s, &idx);
    }
    displayProduct(s, &idx);

    int srchId;
    printf("\nEnter product id you want to search: ");
    scanf("%d", &srchId);
    int x = searchByAddress(s, srchId, &idx);
    printByAddr(&s[x]);
}
```

Output:

```
PS C:\Code>  & 'c:\Users\bhagv\.vscode\..\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter ID:76
Enter Name of product:XYZ
Enter quantiy:1
Enter Price:45000
ID: 76
```

```
Name: XYZ
Quantiy: 1
Price: 45000
How many Products do you want to Store : 2
Enter ID:23
Enter Name of product:jhvahc
Enter quantiy:324
Enter Price:43435
Enter ID:54
Enter Name of product:jkbd
Enter quantiy:33
Enter Price:87478289

 Entered data
ID: 23
Name: jhvahc
Quantiy: 324
Price: 43435

ID: 54
Name: jkbd
Quantiy: 33
Price: 87478289
Enter product id you want to search: 23

ID: 23
Name: jhvahc
Quantiy: 324
Price: 43435
PS C:\Code>
```