UNIT 7

Structure and Union

Structure

- Structure is a collection of different or similar data types variable under a common variable name.
- Structure is a convenient tool for handling a group of logically related data items.

Why structure is needed?

Suppose we need to store the data of students like name, address, class, rollno, age etc.

One way of doing this would be creating a different variable name, age, address to store this information separately.

However, when we need to store the data of multiple students, in that case we would need to create these several variables again for each student. This is such a big headache to store data in this way.

We can solve this problem easily by using structure. We can create a structure that has members name, address, class, rollno, age etc. and then we can create the variables of this structure or each student.

Example:

```
struct student
{
  char name[20];
  char address[20];
  int class;
  int rollno;
  int age;
}
```

Now multiple variable of struct student type can be declared as:

```
int main()
{
 struct student st1, st2,st3. . . . stn;
 . . . . . .
 return 0;
}
```

Defining structure

Structure must be defined first for their format that may be used later to declare structure variables. The general format or syntax to create structure is:

```
struct   structure_name
 {
        data_type    member1;
        data_type     member2;
        . . . . . .
        . . . . . .
        data_type     membern;
};
```

For example

```
struct student
{
  char name[20];
  int roll;
  float marks;
};
```

Here a derived type struct student is defined. Student is a structure name and name, roll, marks, are called structure elements or members. Each of these members belongs to different data types.

Declaration of structure variables

Method I	

Syntax:
struct structure_name
{
datatype member1;
datatype member2;
. . . . . .
datatype memebern;
};
int main()
{
struct structure_name variable1,
variable2,……..variable n;
. . . . . .
return 0;
}
	Example
	struct student
	{
	char name[20];
	int roll;
	float marks;
	};
	int main()
	{
	struct student st1, st2, st3;
	}

Method II	
Syntax: struct structure_name { datatype member1; datatype member2; . . . . . . datatype memebern; } variable1, variable2, ... variable n;	***Example*** struct student { char name[20]; int roll; float marks; } st1,st2, st3;

Accessing Member of Structure

How members of the structure are accessed? Show it with example.

To access any member of a structure, we use the dot operator (.) .

Syntax: **structure_variable.member**

Here structure variable refers to the name of structure type variable and member refers to name of member within structure. For example, consider the following structure

```
struct student
{
char name[20];
int roll;
float marks;
};
  int main()
{
struct student st;
. . . . . .. . . . . . . . . .
}
```

Now,

- ✓ name of st is given by st.name
- ✓ roll of st is given by st.roll
- ✓ marks of st is given by st.marks

We can use st.name, st.roll, st.marks etc. like other ordinary variables in the program. They can be read, displayed, processed, assigned values or can be send to functions as arguments.

Program:

```
#include<stdio.h>
#include<string.h>
struct student
{
      char name[20];
      int roll;
      float marks;
  };
int main()
{
    struct student st;
    strcpy(st.name,"Ram");
    st.roll=15;
    st.marks=85.5;
    printf("Name of student is %s\n",st.name);
     printf("Roll number= %d\n",st.roll);
     printf("Marks obtained=%f\n",st.marks);
     return 0;
}
```

Output
Name of student is Ram
Roll number=15
Marks Obtained=85.500000

Initialization of structure

Initialization of structure at compile time

Structure variable can be also initialized at compile time. The values to be initialized must appear in order as in the definition of structure within braces and separated by commas.

Syntax:
struct structure_name structure_variable={value1,value2, ..., value n };

where value1 is initialized to the first member, value2 is initialized to second member and so on.

For example:
If the structure is declared as

```
struct student
{
    char name[20];
    int rollno;
    float marks;
}
```

The variable of this type can be initialized during its declaration as shown below.

```
struct student st={"Ram",100,56.5};
```

This line is equivalent to

```
struct student st;
strcpy(st.name,"Ram");
st.rollno=15;
st.marks=56.5;
```

Program:

```
#include<stdio.h>
struct student
{
  char name[20];
  int roll;
  float marks;
};
```

```
int main()
{
  struct student st={"Ram",15,85.5};
  printf("Name of student is %s\n",st.name);
  printf("Roll number= %d\n",st.roll);
  printf("Marks obtained=%f\n",st.marks);
  return 0;
}
```

Initialization of structure at runtime

```
#include<stdio.h>
struct student
{
   char name[20];
   int roll;
   float marks;
};
int main()
{
  struct student st;
  printf("Enter the name of student\n");
  gets(st.name);
  printf("Enter the Roll number of student\n");
  scanf("%d",&st.roll);
  printf("Enter the marks of student\n");
  scanf("%f",&st.marks);
  printf("Name of student is\n");
  puts(st.name);
  printf("Roll number= %d\n",st.roll);
  printf("Marks obtained=%f\n",st.marks);
  return 0;
}
```

Nested structure

The structure within structure is called Nested structure. In other words the individual members of structure can be other structure as well.

Let us consider structure date which has members day, month and year. This structure can be nested another structure say employee.ie. structure date is member of another structure employee.

Eg.

```
struct date
{
    int day;
    int month;
    int year;
};
```

This structure can be nested within another structure as its member.

```
 structure employee
{
   char name[50];
   int id;
   struct date dob;
   float salary;
}e1;
```

Accessing member of nested structure

How the members of the nested structure are accessed? Show it with example.

In general member of nested structure is accessed as:

structure_variable.member.submember;

Here, we can access the member of above nested structure as:

e1.dob.year→ year of birthdate of e1
e1.dob.month→ month of birthdate of e1
e1.dob.day → day of birthdate of e1

Program to illustrate accessing the nested structure components

```
 #include<stdio.h>
struct date
{
int year;
int month;
int day;
};

struct employee
{
char name[20];
int id;
struct date dob;
float salary;
}e1;
```

```
int main()
{
printf("Enter the name of employee\n");
scanf("%s",e1.name);
printf("Enter ID of employee\n");
scanf("%d",&e1.id);
printf("Enter the year of birthday\n");
scanf("%d",&e1.dob.year);
printf("Enter the month of birthday\n");
scanf("%d",&e1.dob.month);
printf("Enter the day of birthday\n");
scanf("%d",&e1.dob.day);
printf("Enter the salary of employee\n");
scanf("%f",&e1.salary);
printf("Detail information of employee is\n");
printf("Name\tId\tDate of birth\tsalary\n");
printf("%s\t%d\t%d-%d-%d\t%f",e1.name,e1.id,e1.dob.year,e1.dob.month,e1.dob.day,e1.salary);
return 0;
}
```

Array of structure

Array of structure is the collection of multiple structures variables where each variables contain information about different entities.

Let us consider the structure employee

```
struct employee
{
  char name[20];
  int id;
  float salary;
};
```

Normally, when we want to store the record of multiple employee ,Let's suppose 10 ,we have to declare 10 different structure variables such as e1,e2,e3……. e9, e10.

By using array of structure we can simply declared in following ways;

First way

```
struct employee
{
   char name[20];
    int id;
   float salary;
};
int main()
{
struct employee e[10];

}
```

Second way

```
struct employee
{
   char name[20];
    int id;
   float salary;
} e[10];
```

Example:

Array of structures that stores information of 10 employees and display it.

```
#include<stdio.h>
struct employee
{
char name[20];
int id;
float salary;
};
int main()
{
int i;
struct employee e[10];
for(i=0;i<10;i++)
{
printf("Enter information of employee %d \n",i+1);
printf("Enter the Employee id\n");
scanf("%d",&e[i].id);
printf("Enter the name of employee\n");
gets(e[i].name);
printf("Enter the salary of employee\n");
scanf("%f",&e[i].salary);
}
for(i=0;i<10;i++)
{
printf("Information of employee %d \n",i+1);
printf("Employee id=%d\n",e[i].id);
printf("Name= %s\n",e[i].name);
printf("Salary=%f\n",e[i].salary);
}
return 0;
}
```

How to declare and initialize array of structure variables?

Suppose we want declare a structure to store 5 employee details. Array of structure to store 5 employee details is can be declared as follows:

```
struct employee
{
  char name[20];
   int id;
  float salary;
};
```

```
int main()
{
struct employee e[5];

}
```

Structure array initialization follows same syntax as we initialize single structure variable. The only difference is here we can initialize more than one structure variable at a time.

```
struct employee e[5] =

{

   { "Ramesh", 12,15000.35},

   { "Hari", 15, 25000.5 },

   { "Gopal",  17, 30000.25 },

   { "Sita", 2, 45000.56 },

   { "Roshan",  9, 32000.25}

};
```

Program

```
#include<stdio.h>
struct employee
{
char name[20];
int id;
float salary;
};
int main()
{
int i;
struct employee e[5] =
{
{ "Ramesh",1,15000.35},
{ "Hari",2, 25000.5 },
{ "Gopal", 3, 30000.25 },
{ "Sita", 4,45000.56 },
{ "Roshan",5, 32000.25}
};
printf("Employee Information List:\n");
printf("ID\tName\tSalary\n");
for(i=0;i<5;i++)
{
printf("%d\t%s\t%f\n",e[i].id,e[i].name,e[i].salary);
}
return 0;
}
```

Array within Structure

We can use array of any type as the member of structure according to our programming requirement. Array within structure can be of single structure or of an array of structures. The following program shows a structure definition having a float type array to read marks of 5 subjects of student and display the average marks.

```
#include<stdio.h>
struct student
{
        char name[20];
        int roll;
        float marks[5];
};
int main()
{
 struct student st;
 int i;
 float avg,sum=0;
 printf("Enter the name of student\n");
 gets(st.name);
 printf("Enter the roll no of student\n");
 scanf("%d",&st.roll);
 for(i=0;i<5;i++)
 {
 printf("Enter marks of subject %d\n",i+1);
 scanf("%f",&st.marks[i]);
 sum=sum+st.marks[i];
 }
 avg=sum/5;
 printf("Informatin of student  is\n");
 printf("Name:%s\tRoll:%d\tAverage marks:%f",st.name,st.roll,avg);
 return 0;
}
```

Structures and Pointers

Pointers can also be used pointing to a structure.

We can define pointers to structures in the following way

struct structure_name *pointer_name;

We can store the address of a structure variable in following way:

pointer_name = &structure_variable name;

To access the members of a structure using a pointer to that structure, you must use the → operator as follows –

pointer_name->member;

Program

```
#include<stdio.h>
struct book
{
 char name[30];
 int pages;
 float price;
};
 int main()
 {
        struct book b={"Programming in C",300,550.35};
        struct book *ptr;
        ptr=&b;
        printf("Book name=%s\n",ptr->name);
        printf("Number of pages=%d\n",ptr->pages);
        printf("Book price=%f\n",ptr->price);
        return 0;
}
```

Self-referential structure

Write a short notes on: Self-referential structure.

The self-referential structure is a structure that contain pointer to a structure of the same type.

General form:	Example

struct structure_name
{
datatype member1;
datatype member2;
. . . . . .
 	
struct structure_name *pointer_name;
};
	struct node
	{
	
	int data;
	
	struct node * next;
	
	};

Here the structure node is referencing to self as it has a pointer of its own type 'struct node', named 'next'.
Such self-referential structures are very useful in applications that involve linked data structures, such as lists and trees.

```
#include<stdio.h>
struct node
{
        int data;
        struct node *next;
};
```

```
int main()
{
        struct node a,b,c;
        a.data=10;
        b.data=20;
        c.data=30;
        a.next=&b;
        b.next=&c;
        c.next=NULL;
        printf("a=%d\n",a.data);
        printf("b=%d\n",a.next->data);
        printf("c=%d",b.next->data);
        return 0;
}
```

Output:

```
a=10
b=20
c=30
```

Union

Union is a derived data type which allows to store number of variables of different data types in same memory location.

Unions are similar to structure but there is only difference in terms of storage. In structures, each member has its own memory location but all member of the union use the same storage location. Due to sharing of a common memory location, all the members of union cannot accessed at a time.

The memory occupied by the union will be the memory sized occupied by a member which requires largest memory space among members.

Eg.

```
union student
 {
    char name[20];
    int roll;
    float marks;
};
```

Here, memory required for member name is 20 bytes, roll is 2 bytes and marks is 4 bytes. Here, largest memory space (ie.20 bytes) will be allocated for union.

Declaration of union

Method I	

Syntax:
union union_name
{
datatype member1;
datatype member2;
……..
datatype memebern;
};
int main()
{
union union_name variable1, variable2,……..variable n;
}
	Example:
	union student
	{
	char name[20];
	int roll;
	float marks;
	};
	int main()
	{
	union student st1, st2, st3;
	}

Method II	
Syntax: union union_name { datatype member1; datatype member2; …….. datatype memebern; } variable1, variable2,……..variable n;	***Example:*** union student { char name[20]; int roll; float marks; } st1,st2, st3;

Example:

```
#include<stdio.h>
union student
{
        char name[20];
        int roll;
        float marks;
};
int main()
{
        union student st;
        printf("Enter the Name\n");
        scanf("%s",st.name);
        printf("Name is:%s\n",st.name);
        printf("Enter the Rollno\n");
        scanf("%d",&st.roll);
        printf("Rollno is %d\n",st.roll);
        printf("Enter the marks\n");
        scanf("%f",&st.marks);
        printf("Marks is %f",st.marks);
        return 0;
}
```

How is structure different from union? Give examples codes.

Both structure and union are used to store the number of variables of different data types under a common variable name.

The main difference among them is in terms of storage. In structure each member has its own storage location but all members of union use the common memory location. Due to sharing of common memory location all members cannot be accessed at a time.

Program to illustrate Structure	**Program to illustrate Union**

#include<stdio.h>
struct student
{
  int roll;
  float marks;
};
int main()
{
  struct student st;
  st.roll=15;
  st.marks=67.5;
  printf("Roll number= %d\n",st.roll);
  printf("Marks=%f",st.marks);
  return 0;
}
 	
Output:
Roll number=15
Marks=67.5
	#include<stdio.h>
	union student
	{
	  int roll;
	  float marks;
	};
	int main()
	{
	  union student st;
	  st.roll=15;
	  st.marks=67.5;
	  printf("Roll number=%d\n",st.roll);
	  printf("Marks=%f",st.marks);
	  return 0;
	}
	
	Output:
	Roll number=***(Garbage value)***
	Marks=67.5

Description: Here the memory reserved for structure will be (2 bytes+4 bytes=6 bytes) 2 bytes is reserved for integer roll and 4 bytes is reserved for float marks As well all members can be used at the same time.	**Description:** Here the memory reserved for union will be 4 bytes because data type of marks is float which is the largest member of union. But, all members cannot be used at the same time because all members in union share the common memory location.

Write a short notes on:
Differences between structure and union.

Structure	**Union**
Keyword **struct** is used.	Keyword **union** is used.

Syntax
struct structure_name
{
datatype member1;
datatype member2;
. . . . . .
datatype membern;
} ;
	Syntax:
	union union_name
	{
	datatype member1;
	datatype member2;
	. . . . . .
	datatype membern;
	};

The separate memory location is allocated to each member of the **structure**.	All members of the **union** share the same memory location.
All the members of structure can be accessed at a time.	Only one members of union can be accessed at a time.
Size of the structure is equal to the sum of size of the each member.	Size of the union is equal to the size of the largest member.

Eg.
struct employee
{
char name[20];
int age;
float salary;
};
Memory reserved=(20+2+4)=26 bytes
	Eg.
	union employee
	{
	char name[20];
	int age;
	float salary;
	};
	Memory reserved=20 bytes

Some Important Questions:

1. Differentiate structure and union.
2. Define structure and union. Explain way of declaring and accessing member of them with suitable example.
3. Compare and contrast structure with union.

Create a structure called employee having name, address, salary and age. Input n records of employee and display information of employee whose address is Kathmandu.

```
#include<stdio.h>
#include<string.h>
struct employee
{
        char name[20];
        char address[20];
        float salary;
        int age;
 };
int main()
{
 struct employee e[100];
 int i,n;
 printf("Enter how many records you want to enter\n");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
         printf("Enter the records of employee %d\n",i+1);
         printf("Enter the name\n");
         fflush(stdin);
         gets(e[i].name);
         printf("Enter the address\n");
         gets(e[i].address);
         printf("Enter the salary\n");
         scanf("%f",&e[i].salary);
         printf("Enter the age\n");
        scanf("%d",&e[i].age);
 }
 printf("Information of employee whose address is kathmandu are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
   if(strcmp(e[i].address,"kathmandu")==0)
   {
   printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address, e[i].salary,e[i].age);
   }
 }
 return 0;
}
```

Perform the following operations for above question:

Display information of employee whose name is raju

Hint:

```
printf("Information of student whose name is raju are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
   if(strcmp(e[i].name,"raju")==0)
   {
   printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address, e[i].salary,e[i].age);
   }
 }
```

Display information of employee whose salary is greater than 10, 000

Hint:

```
printf("Information of employee whose salary is greater than 10,000 are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
   if(e[i].salary>10000)
   {
   printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address,e[i].salary,e[i].age);
   }
 }
```

Display information of employee whose age is greater than 30

Hint:

```
printf("Information of employee whose age is greater than 30 are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
   if(e[i].age>30)
   {
   printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address,e[i].salary,e[i].age);
   }
}
```

Display the information of Employee who has highest salary

Hint:

```
float high;
. . . . . .
. . . . . . . . . . . . . .
high=e[0].salary;
 for(i=0;i<n;i++)
 {
        if(e[i].salary>high)
        {
          high=e[i].salary;
        }
 }
 printf("Information of employee who have highest salary are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
        if(e[i].salary==high)
        {
         printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address,e[i].salary,e[i].age);
        }
}
```

Display all the information of employee

Hint:

```
printf("Information of employee are\n");
 printf("Name\tAddress\tSalary\tAge\n");
 for(i=0;i<n;i++)
 {
   printf("%s\t%s\t%f\t%d\n",e[i].name,e[i].address,e[i].salary,e[i].age);
 }
```

Assignment:

1) Create a structure called customer having name, address, account number and balance. Input n records and
   - Display information of customer whose balance is greater than 10000
   - Ask for account number from customer and display their information
     *[Note: Data type for accountno will be **long int** and format specifier **%ld**]*
2) Create a structure called employee having name, address, post salary and age. Input n records of employee and display information of those employee whose post is accountant.
3) Create a structure called university having name, college, and number of faculties. Input n records of college and display the name of those colleges whose address is Kathmandu.

4) Create a structure to specify students data rollno,name,address,marks1,mark2,mark3 to read 20 students data into array of structures and print the information of those students who are failed.
(Note: Pass mark for any subject is 45)

1) Create a structure for the following data.

Roll.no	**Name**	**Address**	**Faculty**	**Data of birth**		
				mm	**dd**	**yy**

Write a program to input 100 students and display the records of student of "computer" faculty.

```
#include<stdio.h>
#include<string.h>
struct dob
{
        int dd;
        int mm;
        int yy;
};
struct student
{
        int roll;
        char name[20];
        char address[20];
        char faculty[20];
        struct dob d;

};
int main()
{
 int i;
 struct student st[100];
 for(i=0;i<100;i++)
 {
   printf("Enter the records of student %d\n",i+1);
   printf("Enter the roll number\n");
   scanf("%d",&st[i].roll);
   printf("Enter the name\n");
   gets(st[i].name);
   printf("Enter the address\n");
   gets(st[i].address);
   printf("Enter the faculty\n");
   gets(st[i].faculty);
```

```
  printf("Enter year of birthday\n");
  scanf("%d",&st[i].d.yy);
  printf("Enter month of birthday\n");
  scanf("%d",&st[i].d.mm);
  printf("Enter day of birthday\n");
  scanf("%d",&st[i].d.dd);
 }
 printf("The records of student whose faculty is computer are\n");
 printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
 for(i=0;i<100;i++)
 {
  if(strcmp(st[i].faculty,"computer")==0)
  {
  printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
  st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
  }
 }
 return 0;
}
```

Perform the following operations for above questions.

Display the records of student whose address is “pokhara”

Hint:

```
printf("The records of student whose address is pokhara are\n");
printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
for(i=0;i<100;i++)
 {
  if(strcmp(st[i].address,"pokhara")==0)
  {
  printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
  st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
  }
 }
```

Display the records of student who are not from “kathmandu”

Hint:

```
printf("The records of student who are not from kathmandu are\n");
printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
for(i=0;i<100;i++)
 {
  if(strcmp(st[i].address,"kathmandu")!=0)
  {
  printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
  st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
  }
 }
```

Display the records of student whose name is “raju”

Hint:

```
printf("The records of student whose name is raju are\n");
printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
for(i=0;i<100;i++)
 {
  if(strcmp(st[i].name,"raju")==0)
  {
  printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
  st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
  }
}
```

Display records of student who born in july month

Hint:

```
printf("The records of student who born in  july are\n");
printf("\nRollno\tName\tAddress\tFaculty\tDate of Birth");
   for(i=0;i<100;i++)
   {
          if(st[i].d.mm==7)
           {
           printf("\n%d\t%s\t%s\t%s\t%d\t%d\t%d",st[i].roll,st[i].name,st[i].address,
           st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
          }
   }
```

Display records of student who was born in 1995

Hint:

```
printf("The records of student who was born in  1995 are\n");
printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
   for(i=0;i<100;i++)
  {
          if(st[i].d.yy==1995)
           {
           printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
           st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
          }
 }
```

Display all the records

Hint:

```
printf("The records of all student are\n");
 printf("Rollno\tName\tAddress\tFaculty\tDate of Birth\n");
 for(i=0;i<100;i++)
 {
          printf("%d\t%s\t%s\t%s\t%d\t%d\t%d\n",st[i].roll,st[i].name,st[i].address,
         st[i].faculty,st[i].d.mm,st[i].d.dd,st[i].d.yy);
}
```

Create a structure called book, member name, price, author, and published date in day, month, and year. Write a program to read 100 books information from the user and displays those records having price greater than 250.

```
#include<stdio.h>
struct date
{
        int dd;
        int mm;
        int yy;
};
struct book
{
        char name[20];
        float price;
        char author[20];
        struct date d;
};
int main()
{
 int i;
 struct book b[100];
 for(i=0;i<100;i++)
 {
  printf("Enter the information of book %d\n",i+1);
  printf("Enter the book name\n");
  gets(b[i].name);
  printf("Enter the author name\n");
  gets(b[i].author);
  printf("Enter the book price\n");
  scanf("%f",&b[i].price);
  printf("Enter book published year\n");
  scanf("%d",&b[i].d.yy);
  printf("Enter book published month\n");
  scanf("%d",&b[i].d.mm);
  printf("Enter book published date\n");
  scanf("%d",&b[i].d.dd);
 }
 printf("Information of Books having price greater than 250 are\n");
 printf("Name\tPrice\tAuthor\tPublished date\n");
 for(i=0;i<100;i++)
 {
        if(b[i].price>250)
        {
   printf("%s\t%f\t%s\t%d\t%d\t%d\n",b[i].name,b[i].price,b[i].author,b[i].d.dd,b[i].d.mm,
   b[i].d.yy);
        }
 }
 return 0;  }
```

Assignment.

1. Write a Program to input the following records of 50 employees using structure and display them properly.

Name	Address	Post	Salary	Data of Appointment		
				mm	dd	yy

2. Given a structure of employee

Name	Address	telephone	Salary	Year of joining		
				mm	dd	yy

Write a program to input data of 100 employee and display the record of those employees living in pokhara.

*[Note: datatype for telephone must be **long int** and format specifier is **%ld**]*

3. Create a structure for the following data.

Emp_id	Emp_name	Address	Department	Date of Birth		
				mm	dd	yy

Also write a program to input 100 employee records and display whose Department is “sales”.

4. Create a structure called student with data members name, rollno and college for 10 students. Display the name of students who study at SOE.

5. Write a program to read the records of n students (roll no, name, per, address) using structures and display the records of those student whose address in Chitwan.

6. Create a structure named student. Student has attributes name, address, date of birth and marks in percentage. Input the records for ‘n’ students and display the information of the students who have scored above 75 percentage.

Define a structure called 'football' that will describe the following information:
Player name
Country name
Number of goals scored.

Using football, declare an array player with 50 elements and write a program to read the information about all the information about all the 50 players and print a country wise list containing names of players with their number of goal scored.

```
#include<stdio.h>
#include<string.h>
#define N 50
struct football
 {
   char name[20];
   char country[20];
   int goal;
};
int main()
 {
   struct football p[50],temp;
   int i, j;
   for (i = 0; i < N; i++)
    {
      printf("Enter the infomation of player %d\n",i+1);
      printf("Enter player name\n");
      gets(p[i].name);
      printf("Enter player country\n");
      gets(p[i].country);
      printf("Enter Number of goal scored\n ");
      scanf("%d", &p[i].goal);
   }
  for (i = 0; i < N-1; i++)
  {
    for (j = 0; j < N - 1-i; j++)
            {
           if (strcmp(p[j].country, p[j + 1].country) > 0)
                      {
          temp = p[j];
          p[j] = p[j + 1];
          p[j + 1] = temp;
        }
     }
  }
```

```
  printf("Country wise list of players are \n");
  printf("Player name\tCountry\tNo of goal scored\n");
  for (i = 0;i < N;i++)
  {
    printf("%s\t%s\t%d\n",p[i].name,p[i].country,p[i].goal);
  }
}
```

Assignment:

Define a structure called cricket that will describe the following information:
player name,
team name,
batting average.
Using cricket, declare an array player with 50 elements and write a program to read the information about all the 50 players and print a team-wise list containing names of player with their batting average.

In a bank there are n customers with attributes name, account no and balance. Write a Program to find out information of customer who has highest balance in bank.

```
#include<stdio.h>
struct customer
{
        char name[20];
        long int accno;
        float balance;
};

int main()
{
 int i,n;
 float high;
 struct customer c[100];
 printf("Enter the number of customer\n");
 scanf("%d",&n);
```

```
printf("Enter records of %d customer\n",n);
for(i=0;i<n;i++)
{
  printf("Enter Name\n");
  gets(c[i].name);
  printf("Enter account no\n");
  scanf("%ld",&c[i].accno);
  printf("Enter balance\n");
  scanf("%f",&c[i].balance);
}

high=c[0].balance;
for(i=0;i<n;i++)
{
  if(c[i].balance>high)
  {
    high=c[i].balance;
  }
}
printf("Information of customer who have highest balance are\n");
printf("Name\tAccount no\tBalance\n");
for(i=0;i<n;i++)
{
 if(c[i].balance==high)
 {
          printf("%s\t%ld\t%f\n",c[i].name,c[i].accno,c[i].balance);
 }
}
 return 0;
}
```

Assignment:

Write a C program to accept details of 'n' employee (eno, ename, salary) and display the details of employee having highest salary. Use array of structure.

Create a structure called student with data members name, rollno, and marks of three subjects for 100 students. Display the name of students with average marks greater than 80.

```
#include<stdio.h>
struct student
{
        char name[20];
        int rollno;
        float  sub1,sub2,sub3;
        float avg;
};
int main()
{
 int i;
 struct student st[100];
 for(i=0;i<100;i++)
 {
   printf("Enter information of student %d\n",i+1);
   printf("Enter Name\n");
   gets(st[i].name);
   printf("Enter Rollno\n");
   scanf("%d",&st[i].rollno);
   printf("Enter marks of sub1\n");
   scanf("%f",&st[i].sub1);
   printf("Enter marks of sub2\n");
   scanf("%f",&st[i].sub2);
   printf("Enter marks of sub3\n");
   scanf("%f",&st[i].sub3);
   st[i].avg=(st[i].sub1+st[i].sub2+st[i].sub3)/3;
 }
 printf("Name of student with average marks greater than 80\n");
 for(i=0;i<100;i++)
 {
   if(st[i].avg>80)
   {
     puts(st[i].name);
   }
 }
 return 0;
}
```