20)Describe the concept of pointer with function using an example.

Pointer to Function

- Every function has reference or address, and if we know the reference or address of function, we can access the function using its reference or address.
- This is the way of accessing function using pointer.

Syntax

```
return-type (*ptr-function)(argument list);
```

- return-type: Type of value function will return.
- argument list: Represents the type and number of value function will take, values are sent by the calling statement.
- (*ptr-function): The parentheses around *ptr-function tells the compiler that it is pointer to function.
- If we write *ptr-function without parentheses then it tells the compiler that ptr-function is a function that will return a pointer.

17

Write a program to sum of two numbers using pointer to function.

Program

```
#include<stdio.h>
int Sum(int,int);
int (*ptr)(int,int);
int main()
{
    int a,b,rt;
    printf("\nEnter 1st number : ");
    scanf("%d",&a);
    printf("\nEnter 2nd number : ");
    scanf("%d",&b);
    ptr = Sum;
    rt = (*ptr)(a,b);
    printf("\nThe sum is : %d",rt);
    return 0;
}
int Sum(int x,int y)
{
    return x + y;
}
```

Output

```
Enter 1st number : 5

Enter 2nd number : 10

The sum is : 15
```

21)Define and describe structure.

What is Structure?

- Structure is a collection of logically related data items of different datatypes grouped together under single name.
- Structure is a user defined datatype.
- Structure helps to build a complex datatype which is more meaningful than an array.
- But, an array holds similar datatype record, when structure holds different datatypes records.
- Two fundamental aspects of Structure:
  - Declaration of Structure Variable
  - Accessing of Structure Member

Syntax to Define Structure

- To define a structure, we need to use `struct` keyword.
- This keyword is reserved word in C language. We can only use it for structure and its object declaration.

Syntax

```
1 struct structure_name
2 {
3     member1_declaration;
4     member2_declaration;
5     . . .
6     memberN_declaration;
7 };
```

me of custom type

ember

- Members can be normal variables, pointers, arrays or other structures.
- Member names within the particular structure must be distinct from one another.

Example to Define Structure

Example

```c
struct student
{
    char name[30]; // Student Name
    int roll_no; // Student Roll No
    float CPI; // Student CPI
    int backlog; // Student Backlog
};
```

- You must terminate structure definition with semicolon ;.
- You cannot assign value to members inside the structure definition, it will cause compilation error.

Example

```c
struct student
{
    char name[30] = "ABC"; // Student Name
    . . .
};
```