Pointers

The pointer in C language is a variable which stores the address of another variable. This variable can be of type int, char, array, function, or any other pointer. The size of the pointer depends on the architecture. However, in 32-bit architecture the size of a pointer is 2 byte.

Consider the following example to define a pointer which stores the address of an integer.

1. **int** n = 10;
2. **int*** p = &n; // Variable p of type pointer is pointing to the address of the variable n of type integer.

Declaring a pointer

The pointer in c language can be declared using * (asterisk symbol). It is also known as indirection pointer used to dereference a pointer.

1. **int** *a;//pointer to int
2. **char** *c;//pointer to char

Pointer Example

An example of using pointers to print the address and value is given below.

1. #include<stdio.h>
2. **int** main(){
3. **int** number=50;
4. **int** *p;
5. p=&number;//stores the address of number variable
6. printf("Address of p variable is %x \n",p); // p contains the address of the number therefore printing p gives the address of number.
7. printf("Value of p variable is %d \n",*p); // As we know that * is used to dereference a pointer therefore if we print *p, we will get the value stored at the address contained by p.
8. **return** 0;

9. }

Output

sorting array elements

```c
#include <stdio.h>
void main()
{
int *a,i,j,tmp,n;
        printf("\n\n pointer : sort an array using pointer :\n");
        printf("--------------------------------------------\n");
printf(" input the number of elements to store in the array : ");
scanf("%d",&n);
printf(" input %d number of elements in the array : \n",n);
  for(i=0;i<n;i++)
    {
        printf(" element - %d : ",i+1);
        scanf("%d",a+i);
        }
  for(i=0;i<n;i++)
  {
   for(j=i+1;j<n;j++)
   {
     if( *(a+i) > *(a+j))
     {
tmp = *(a+i);
    *(a+i) = *(a+j);
    *(a+j) = tmp;
     }
    }
    }
printf("\n the elements in the array after sorting : \n");
  for(i=0;i<n;i++)
    {
        printf(" element - %d : %d \n",i+1,*(a+i));
```

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

OUTPUT:

Pointer : Sort an array using pointer :

Input the number of elements to store in the array : 5

Input 5 number of elements in the array :

Element – 1 : 25

Element – 2 : 45

Element – 3 : 89

Element – 4 : 15

Element – 5 : 82

The elements in the array after sorting :

Element – 1 : 15

Element – 2 : 25

Element – 3 : 45

Element – 4 : 82

Element – 5 : 89