**Introduction to pointers**

**Introduction**

Pointer is a variable that stores the address of another variable. Pointer allows you to refer to the same memory space from multiple locations. That means you can update the memory in one location, and the change can be seen from another location in your program.

**How to use pointers?**

* First, we have to define a pointer variable.
* After defining a pointer variable, we can assign the address of a variable to a pointer by using a unary operator (&).
* To access the given variable’s value, we can use the pointer variable with the help of the unary operator (\*). A unary operator is used before the pointer variable to get the value of that address stored in the pointer variable.

**Syntax of pointer**

data\_type \*variable\_name;

Here type is defined as the datatype, and variable\_name is the pointer’s name in which you want to store the address of a given variable.

**Code**

#include <bits/stdc++.h>

using namespace std;

int main()

{

   int data=20;

   int \*ptr; // declaration pointer variable

   ptr = &data;  // assigning the address of given variable to the pointer

   cout<<"Address in ptr="<<ptr<<endl; // gives the address that stored in pointer

   cout<<"Value of data="<<data<<endl; // value of the given variable

   cout<<"Value at \*ptr="<<\*ptr<<endl;  // value of the given variable

   return 0;

}

**Output**

Address in ptr=0x7ffe96baeaec

Value of data=20

Value at \*ptr=20

**Null pointer**

The null pointer is a pointer that does not point to any memory location. It represents an invalid memory location. When we assign a NULL value to a pointer, then that pointer refers to a null pointer.

**Code**

#include <bits/stdc++.h>

using namespace std;

int main()

{

   int \*ptr; // declaration pointer variable

   ptr= NULL;  // assigning NULL Address to the pointer

   cout<<"Address in ptr="<<ptr<<endl; // gives the address that stored in pointer

   return 0;

}

**Output**

Address in ptr=0

**Void pointer**

The pointer with no associated data type is known as a void pointer. A void pointer can hold addresses of any type and can be typecast to any type.

**Code**

#include <bits/stdc++.h>

using namespace std;

int main()

{

   int var=10;

   void \*ptr; // declaration of void pointer variable

   ptr=&var;

   cout<<"Address in ptr="<<ptr<<endl; // gives the address that stored in pointer

   cout<<"Value at ptr="<<\*(int\*)ptr<<endl; // typecating the pointer

   return 0;

}

**Output**

Address in ptr=0x7ffeca383a9c

Value at ptr=10

In the above code, you can see that we have defined a void pointer. This pointer has no associated data type. So, it can point to any type of data. In the ‘cout’ function, you can see that we have first typecasted the pointer into an integer pointer then dereferenced it.

**Array name as a pointer**

The array's name contains the address of the first element of the array, so you can not change the address stored in the array name. The array's name acts as a constant pointer. You can assign the address of an array to a pointer without using an ampersand (&).

**Code**

#include <bits/stdc++.h>

using namespace std;

int main()

{

   int arr[4]={12,15,20,50};

   int \*ptr; // declaration of void pointer variable

   ptr=arr; // assign the address of arr[0] to ptr

   for(int i=0;i<4;i++)

   {

       cout<<\*ptr<<endl;

       ptr++;  // increment pointer ptr by 1

   }

   return 0;

}

**Output**

12

15

20

50

With the help of the above example, you can clearly understand how we can print the elements of the array with the help of the pointer.

**Advantages of using Pointers**

Some advantages of using Pointers are given below:

* Pointers are the variables that are used to store the address of other variables in C/C++.
* Pointer allows us to allocate and deallocate the memory dynamically.
* Pointers help us in simplifying the complexity of the program.
* With the help of pointers, we can improve the execution speed of a program.
* With the help of pointers, we can modify and return more than one variable from a function.

**Frequently asked questions**

1. **Which type of pointer is the most convenient way of storing the raw addresses in c programming?**  
   Ans: void pointer is used to store the raw addresses.
2. **What is a null pointer?**  
   Ans: When we assign a NULL value to a pointer, then that pointer refers to a null pointer.
3. **What is a dangling pointer?**  
   Ans: When deletion or deallocation of referencing object takes place without changing the value of the pointer, it is called dangling pointer.
4. **What is a wild pointer?**  
   Ans: A wild pointer is a pointer that is not initialized properly before using it the first time.