PROGRAMMING IN C

Pointers

CONTENTS

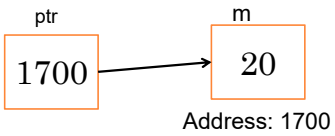
- Introduction to Pointers
- Example
- Features of pointers

POINTERS

- Pointers in C language is a variable that stores/points the address of another variable.
- The pointer variable can be of any data type such as int, float, char, double, short etc.
- Example : **int *p;**
  - **char *q;**
- Where, * is used to denote that "p, q" is pointer variable and not a normal variable.

EXAMPLE

```
void main( )
{
int m=20;
int *ptr;
ptr=&m;
}
```



FEATURES OF POINTERS IN C:

- Normal variable stores the value whereas pointer variable stores the address of the variable.
- The content of the C pointer always be a whole number i.e. address.
- Always C pointer is initialized to null, i.e. int *p = null.
- If a pointer in C is assigned to NULL, it means it is pointing to nothing.
- & symbol is used to get the address of the variable.
- * symbol is used to get the value of the variable that the pointer is pointing to.
- Two pointers of an array can be subtracted to know how many elements are available between these two pointers.
- Pointer addition, multiplication, division are not allowed.

EXAMPLE

```
#include <stdio.h>
void main()
{
  int *ptr, q;
  q = 50;
  /* address of q is assigned to ptr */
  ptr = &q;
  /* display q's value using ptr variable */
  printf("%d", *ptr);
  printf ("%d", ptr);
}
```

Q: The reason for using pointers in a C program is

A: Pointers allow different functions to share and modify their local variables.

B: To pass large structures so that complete copy of the structure can be avoided.

C: Pointers enable complex "linked" data structures like linked lists and binary trees.

D: All of the above

```
#include <stdio.h>
int main()
{
   int arr[] = {1, 2, 3, 4, 5};
   int *p = arr;
   ++*p;
   p += 2;
   printf("%d", *p);
   return 0;
}
```

A: 2
B: 3
C: 4
D: Compiler Error

```
char inchar = 'A';
switch (inchar)
{
case 'A' :
   printf ("choice A n") ;
case 'B' :
   printf ("choice B ") ;
case 'C' :
case 'D' :
case 'E' :
default:
   printf ("No Choice") ;
}
```

A: No choice
B: Choice A
C: Choice A
Choice B No choice
D: Program gives no output as it is erroneous

```
if(a > b)
if(b > c)
s1;
else s2;
```

s2 will be executed if
A: a <= b
B: b > c
C: b >= c and a <= b
D: a > b and b <= c

REFERENCES

- Programming in C, 2011, by J.B. Dixit
- Basics of C Programming, 2011, by J.B. Dixit
- https://www.tutorialspoint.com/cprogramming/c_pointers.htm
- https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/