Pointer

A pointer is a **variable** that stores **memory address**. Like all other variables ,it also **has a name**, **has to be declared** and **occupies some space in memory**. It is called pointer because it points to a **particular location in memory** (by storing the address of that location).

The expression having **(*)** operator is known as **pointer expression**. Pointer is the variable that points to the memory location of the **next variable**. So, we have to assign address of a variable to the pointer variable.

Int a=5;

Int *p1=&a;

If we put *before p1, then, we can access the variable whose address is stored in p1. since p1 contains the address of variable ***a***, we can access the variable ***a*** by writing *p1.

What are Pointers?

Pointers are different from other normal variables. While **other normal variables store values**, pointers are special variables that can hold the address of a variable. Since they store memory address of a variable, the pointers are very commonly said to “point to variables”.

Pointer

- *A pointer is a variable which holds the address of the storage location value for another given variable.*
- C provides two operators & and * which allow pointers to be used in many versatile ways.

*:- is the value of address eg.*a

&:- is the address of variable eg. &a

*(&a):- we are using the value which is inside of the address

pointer arithmetic operations are allowed?

- A pointer can be incremented (++) or decremented (--)
- An integer may be added to a pointer (+ or +=)
- An integer may be subtracted from a pointer (- or -=)
- One pointer may be subtracted from another

In c programming every variable keeps two type of value.

1. Contain of variable or value of variable.
2. Address of variable where it has stored in the memory.

(1) Meaning of following simple pointer declaration and definition:

```
int a=5;
int * ptr;
ptr=&a;
```

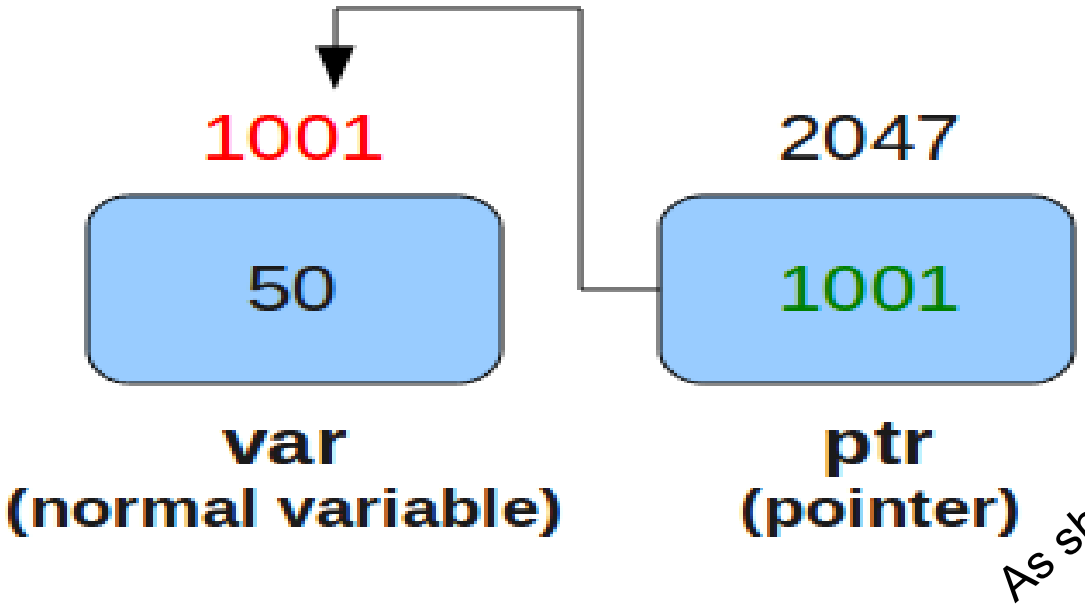
Explanation:

About variable a:

1. Name of variable : a
2. Value of variable which it keeps: 5
3. Address where it has stored in memory : 1025 (assume)

About variable ptr:

4. Name of variable : ptr
5. Value of variable which it keeps: 1025
6. Address where it has stored in memory : 5000 (assume)



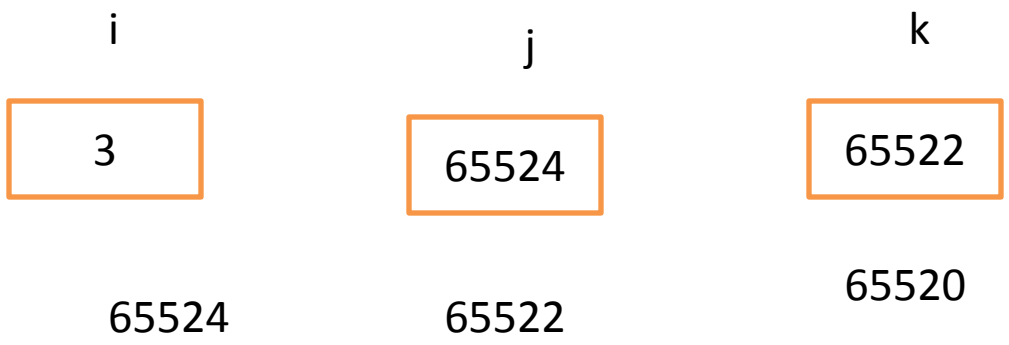
A normal variable ‘var’ has a memory address of 1001 and holds a value 50.

A **pointer variable** has its own address 2047 but stores 1001, which is the address of the variable ‘var’

Another example:

Int I, *j, **K;

Here I is dinary int ,J is a pointer to an int , whereas K is a pointer to an integer pointer



For example

a) valid example

Int *p;

Int num;

P=#

b**) Invalid example**

Int *p;

Float num;

P=# */*as pointer variable p can not store address of float variable*/*

Data type	Conversion Charater
Integer	D
Unsigned integer	U
Octal	O
Hexadecimal	X
Float simple	F
Float exponential	E
Character	C
string	S

```
#include <stdio.h>
#include <conio.h>

 void main( )
{
   int u = 36;
   int *pu; /*pionter to an integer*/
   clrscr();
   pu=&u; /*assign address of u to pu*/
   printf("\nu=%d &u=%x *pu=%d",u,&u,*pu);
   getch();
   }
```

```
/* Illustration of pointer data type */
#include <stdio.h>
 #include <conio.h>
  void main( )
    {
int u = 36;
int v;
int *pu;        /* pointer to an integer*/
int *pv;        /* pointer to an integer*/
clrscr();
pu = &u;        /* assign address of u to pu */
v = *pu;        /* assign address of u to v  */
pv = &v;        /* assign address of v to pv */
printf("\nu = %d \n&u = %x \npu = %x \n*pu = %d", u,&u, pu, *pu);
printf("\n\nv = %d \n&v = %x\n pv = %x \n*pv = %d", v, &v, pv, *pv);
getch();
```

1. WAP to add value stored in two variable using pointer

```
#include <stdio.h>
#include <conio.h>
void main()
  {
 int a,b,sum, *ptra,*ptrb;
 a=20;
 b=10;
 ptra =&a;
 ptrb=&b;
sum=*ptra+*ptrb;
 printf("sum=%d",sum);
 getch();
    }
```