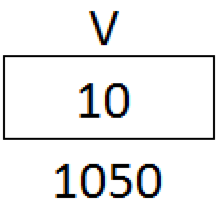
Pointers



A pointer is a variable that represent the location (rather than the value) of a data items.

Suppose v is a variable that represent some particular some particular data item. The compiler will automatically assigns memory cell for this data.

The data item can be access if we know the location of the first memory cell. The address of these memory location can be determined by the expression.

‘&’ operator is called as the address operator that is the address

x=&v   This new variable is called a pointer, since it points the location where it store in memory thus ‘x’ is pointer. In this case value of x=1050.

The data item represented by v i.e. data item stored in these memory cell can be excess by the expression *x, where * is called as indirection operator (or value at address operator).

*x=&v represent the same data value i.e. content of the same memory cell. So *x can easily interchange with the ‘v’

e.g.

```
#include<stdio.h>

int main()

{

        int a,b;

        int v=5;

        int *x;

        a=2*(v+5);

        x=&v;

        b=2*(*x+5);

        printf("z=%d b=%d",a,b);

        return 0;

}
```

e.g. 2

```
#include<stdio.h>

int main()

{

        int v=5;

        int *x;

        x=&v;

        printf("*x=%d v=%d",*x,v);

        *x=0;

        printf("*x=%d v=%d",*x,v);

        return 0;

}
```

Pointer declaration

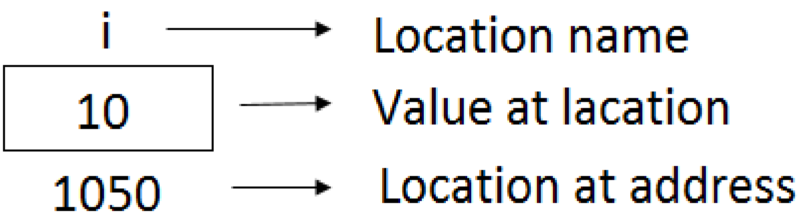
data type *Pointervariable;

Pointer variable like all other variable must be declared before the use. When a pointer value declared, its data type must be same of the data type of the variable, which address it is holding.

```
float a;

float *b;

b=&a;
```

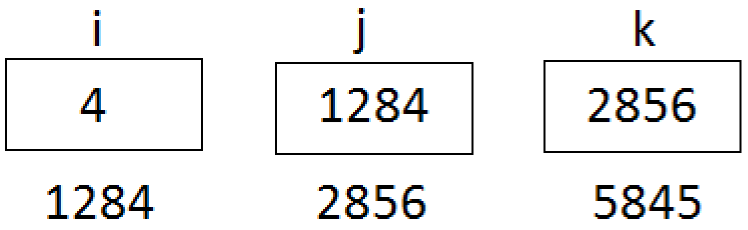


```
Int i, *j, **k;

j=&I;

k=&j;
```

So now *(&i) == *j= = **k == i



Arithmetic operations with pointers

++, --,+,-

Example:

```
#include<stdio.h>
int main()
{
        int a=3;
        int *c=&a;
        int *b=&a;
        printf("c=%d",c);
        c=c+5;
        printf("\nc=%d\n",c);
        c=c-5;
        printf("c=%d\n",c);
        /* c=c+b; Invalid, we cannot add two pointer (address)  */
        int d=*b+*c; /* this is ok as we
                        are adding the values, not the address */
        printf("d=%d\n",d);
        c=++c;
        printf("c=%d\n",c);
        c=--c;
        printf("c=%d\n",c);
        return 0;
}
```