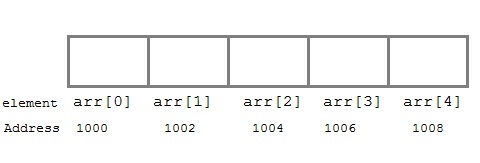
### **Pointer and Arrays**

When an array is declared, compiler allocates sufficient amount of memory to contain all the elements of the array. Base address i.e address of the first element of the array is also allocated by the compiler.

Suppose we declare an array arr,

int arr[5] = { 1, 2, 3, 4, 5 };

Assuming that the base address of arr is 1000 and each integer requires two bytes, the five elements will be stored as follows:



**Here variable arr will give the base address**, which is a constant pointer pointing to the first element of the array, arr[0]. Hence arr contains the address of arr[0] i.e 1000. In short, arr has two purpose - it is the name of the array and it acts as a pointer pointing towards the first element in the array.

**arr is equal to &arr[0] by default**

We can also declare a pointer of type int to point to the array arr.

int \*p;  
p = &arr;   
// or,   
p = &arr[0]; //both the statements are equivalent.

**Now we can access every element of the array arr using p++ to move from one element to another.**

**NOTE:** **You cannot decrement a pointer once incremented. p-- won't work.**

**Pointer to Array**

we can use a pointer to point to an array, and then we can use that pointer to access the array elements. Lets have an example,

#include <stdio.h>

int main()

{

int i;

int a[5] = {1, 2, 3, 4, 5};

int \*p = a; // same as int\*p = &a[0]

for (i = 0; i < 5; i++)

{

printf("%d\n", \*p);

p++;

}

return 0;

}

In the above program, the pointer **\*p(deferencing)** will print all the values stored in the array one by one. We can also use the Base address (a in above case) to act as a pointer and print all the values.

**Pass by reference using pointer**

The value of a variable can be changed in the called function using pointer because they share same address location

#include<stdio.h>

void main()

{

int value = 50;

int \*p;

p = &value;

getPointer(p);

printf("%d\n",value);//value change to 100 - pass by reference

}

void getPointer(int \*q){

printf("%d\n",\*q);

\*q = 100;

}

**Pass by value example**

#include<stdio.h>

void main()

{

int value = 50;

getValue(value);

printf("%d\n",value);

}

void getValue(int v){

v = 100;

}