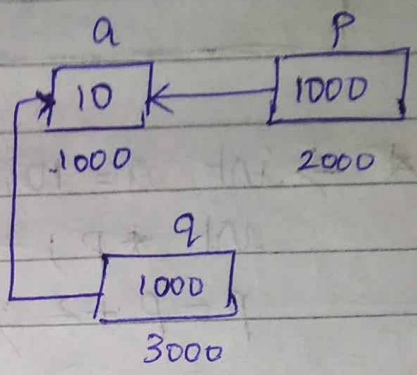
C_75 ⇒ Pointers in C

Pointer Arithmetic (Addition) with program

```
int a = 10;
int *p = &a;
int *q = &a;
```



P+q ✗ wrong

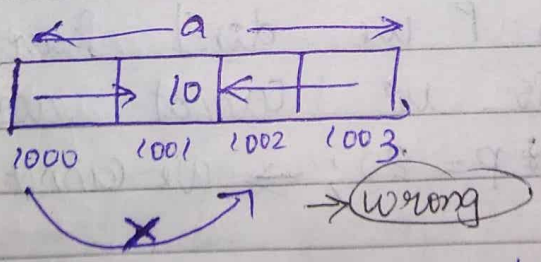
* We cannot perform addition on two pointer variables.

* Instead we can add integer value to a pointer variable.

Eg: p+1, p+2, p+5, q+1, q+5.....

* P = P+2

P = 1000+2 ⇒ 1002 ✗ Wrong



* Pointer is complete address of 'a' (ie) 1000 to 10003

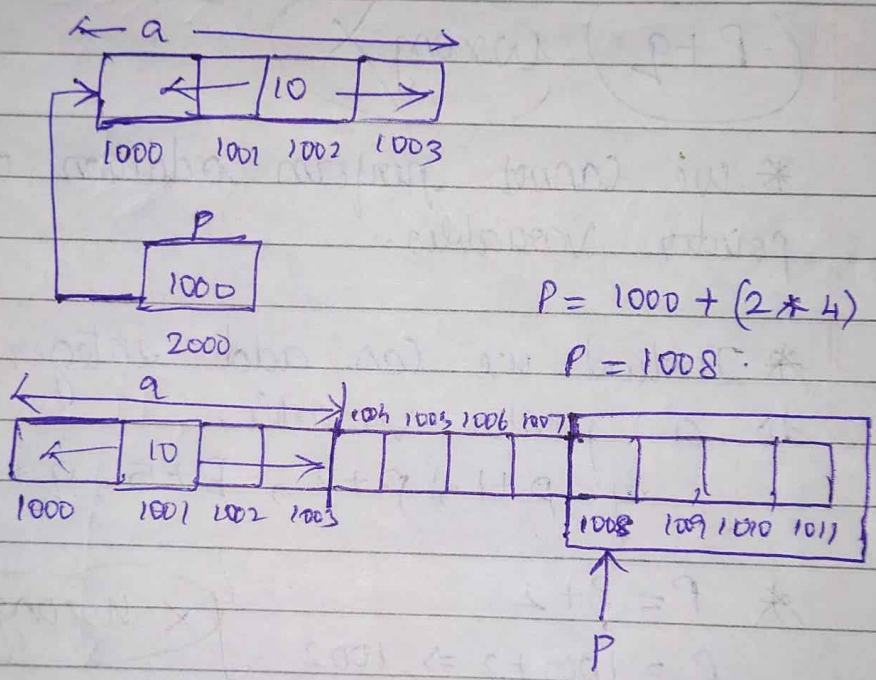
* P = P+1 ⇒ [1000 + (1×4)] ⇒ 1004.

* P = P+2 ⇒ 1000 + (2×4) ⇒ 1008.

* When we add 'n' no/- of integers to the pointer address, formula to get final address of P is;

$$P+n = P+(n * \text{size of int})$$

* → int a = 10;
  int *p; p = &a
  p = p + 2;



→ Now in P we don't have any address initialized; So we cannot assign value
*p = 20; → we cannot do this here

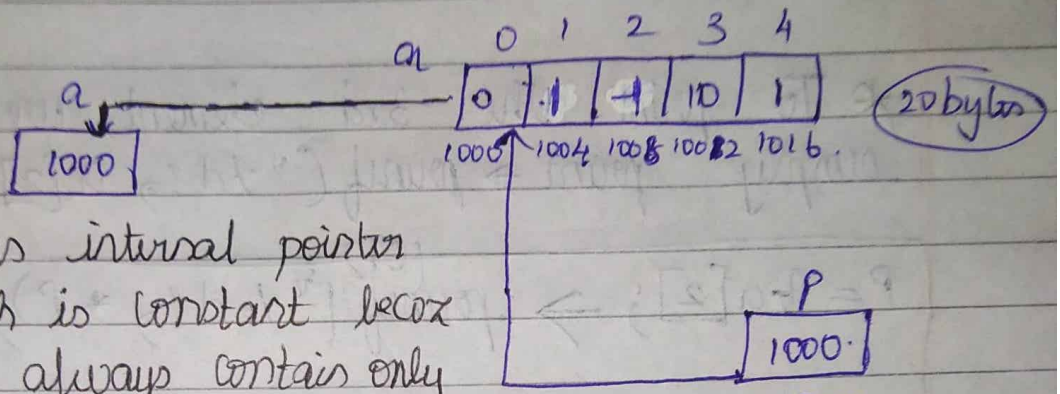
→ printf ("%d", *P);
↳ It will print garbage values.

→ So, it is not useful to deal with only variable.

* Pointer arithmetic will be useful to deal with arrays.

Example:

```
int a[5] = {0,1,-1,10,11};
int *p = &a[0];
```

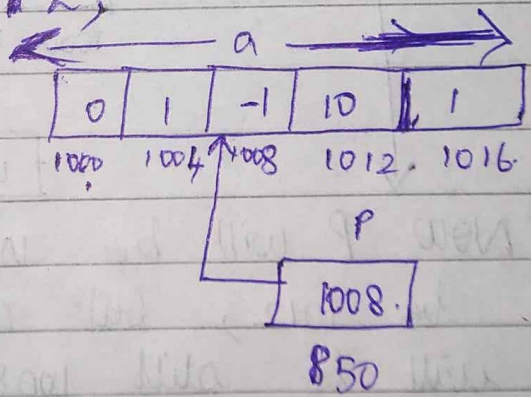


a is internal pointer which is constant because it always contains only base address of 'a'.

p is a pointer which is not constant

```
printf("value of a[0]: %d", *p); ⇒ 0.
```

```
p = p+2;
```




```
printf("value of a[2]: %d", *p);
```

$$p+n = \&a[0+n];$$

$$p+2 = \&a[0+2]$$

$$p+2 = \&a[2]$$

a = a + 2; (wrong)

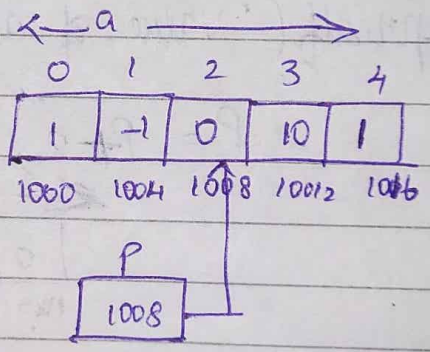
* We cannot move 'a' because a is a internal pointer whose address is always constant which only contains base address.

* To print the 3rd element; we can simply print → printf("%d", a[2]);

P = &a[2]; → printf("%d", *P); (or)

↳ This is using pointer we print the value.

P+1;



* Now P will be increment by one and will be 1012; but not updated in P [1008] it will be still 1008.

* P+1 is equall to P++.

Assignment:

```
int a[5] = {0, 1, -1, 10, 11}
int *P = &a[0];
P = P+1;
*P = 2;
printf("%d", *P); ⇒ 2.
```

CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
/** 1 - POINTER ARTHMETIC ADDITION **/
int main()
{
    //Pointer Addition will be used of Arrays concept
    int a[5]={1,2,3,4,5};

    //To print a[1]
    printf("Value in a[1]:%d\n",a[1]);

    //We can also use pointer variable to print a[1]
    int *p=&a[1];
    printf("Value in a[1]:%d\n",*p);

    //If we want to print a[2] using same pointer variable p
    p=p+1;
    printf("P address in incremented and corresponding address value is:%d\n",*p);
    printf("Address of a[2] is stored in p, Value of a[2]:%d\n",a[2]);
    getch();
}

```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POINTERS\5_P
Value in a[1]:2
Value in a[1]:2
P address in incremented and corresponding address value is:3
Address of a[2] is stored in p, Value of a[2]:3
```

CODE 2:

```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - POINTER ARTHMETIC ADDITION **/
//we have to change the value of 3rd elements with using pointer
//Without using array index; Instead of 3, value should be changed to 10
int main()
{
    int a[5]={1,2,3,4,5};
    int *p=&a[0];
    p=p+2;
    *p=10;
    printf("Value in 3rd element of a[2]:%d %d",*p,a[2]);

    getch();

}
```