C_76 => Pointers in C => Pointer Arithmetic (subtraction) with program

```
int a[] = {0, 1, -1, 10, 11};
```

all same ⇐

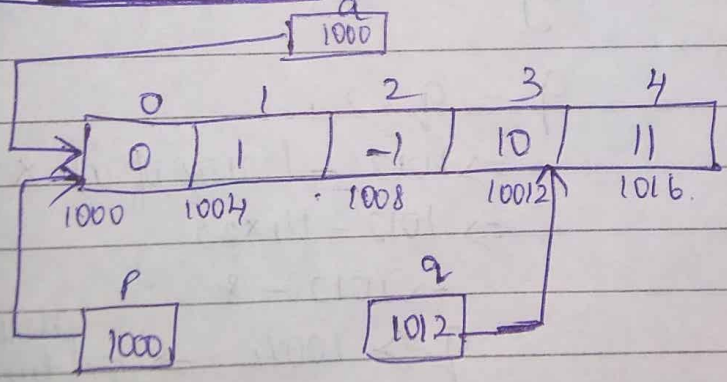
```
int *p = a; // int *p = &a[0]
            // int *p = &a;
```

```
int *p = &a[0];
int *q = &a[3];
```

Note:

* In subtraction P-q is possible ($p-q$ ✓)
* But in addition P+q is not possible ($p+q$ ✗)

$P+2$, $P-2$ is possible.



* a is internal pointer (or) Constant pointer whose address will be always base address of array.

Pointer Subtraction → Middle elements count

* When we subtract two pointers; we get the integer values between two pointers.

```
int d;
```

* d = P - q; ⇒ -4

* d = q - P ⇒ +4

Formula ⇐

$$d \Rightarrow \frac{P - q}{1000 - 1012} \Rightarrow \frac{-12}{4} \Rightarrow -3$$

$$d = \frac{q - P}{1012 - 1000} \Rightarrow \frac{12}{4} \Rightarrow +3$$

size of int

* q = q - 2; → we get address.

// Here pointer moves backward position by 2.

q = q - 2;

⇒ 1012 - (size of int × 2)

⇒ 1012 - (4 × 2)

⇒ 1012 - 8

q ⇒ 1004 ⇒ returns address.

d = P - q;

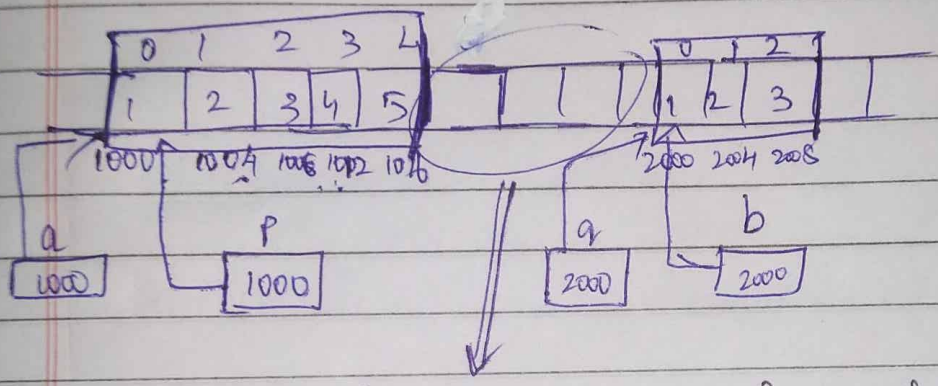
P = P - 1; ⇒ gives garbage value

∴ So better move pointers within array elements.

Hint

* Pointer subtraction can be used only within the same array elements

Eg:
```
int a[5] = {1, 2, 3, 4, 5};
int b[3] = {1, 2, 3};
int *p, *q;   int d;
p = &a;
q = &b;
d = p - q;   ⇒ Here not possible.
```



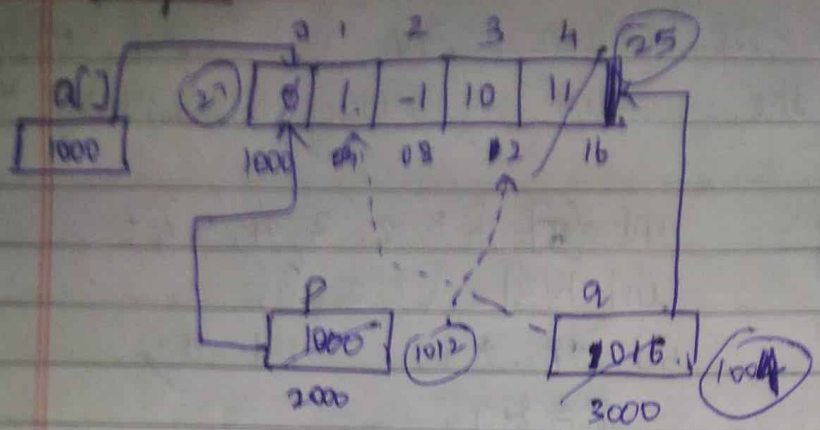
Here we don't know how many bytes of memory occupied between (p and q); so we cannot find the elements between P&q.

Assignment:

```
int a[] = {0, 1, -1, 10, 11};
int *p = &a[0];   // int *p = a;
int *q = &a[4];
int d;
d = p - q;
*q = 25;
d = q - p;
*p = 27;
q = q - 3;
```

```
p = p + 3;
d = p - q;
```

Output:



$d = p - q$

$= 1000 - 1016$

$= 16/4 \Rightarrow 4.$

$d = -4$

$*q = 25.$

Value at (address 1016) = 25

$d = q - p$

$= 1016 - 1000$

$d \Rightarrow 16/4 \Rightarrow 4$

$*p = 27$

value at (address of 1000) = 27

$q = q - 3;$

$q = 1004$

printf ("value at *q: %d", *q);

$3 \times 4 \Rightarrow 12$

1016
12
1004

$d \Rightarrow p - q$

$= 1000 - 1004$

$d = -4/4 \Rightarrow -1$

$P = P + 3;$

$= 1000 + 3$

$= 1000 + (4 \times 3)$

$\Rightarrow 1000 + 12$

$\Rightarrow 1012.$

$1000 + (4 \times 3)$

$\Rightarrow 1000 + 12$

$P \Rightarrow 1012$

printf("Value at address of P : %d", &P);

$\rightarrow$ (10)

$d = P - q;$ $\quad 1012 - 1004 \Rightarrow \frac{8}{4} \rightarrow (2)$

CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
/** 1 - POINTER SUBTRACTION p-q IS POSSIBLE **/
// Pointer addition and subtraction is useful for arrays
//POINTER ADDITION P+Q IS NOT POSSIBLE
//POINTER WITH INTEGER ADDITION OR SUBTRACTION p+1, p-1 IS POSSIBLE
int main()
{
int a[]={0,1,-1,10,11};
int *p=&a[0],*q=&a[4];
int d;
d=p-q;
printf("Value of d:%d\n",d);
*q=25;
d=q-p;
printf("Value of d:%d\n",d);
*p=27;
q=q-3;
printf("Value at address in q:%d\n",*q);
d=p-q;
printf("Value of d:%d\n",d);
p=p+3;
printf("Value at address in p:%d\n",*p);
d=p-q;
printf("Value of d:%d\n",d);
getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POINTE
Value of d:-4
Value of d:4
Value at address in q:1
Value of d:-1
Value at address in p:10
Value of d:2
```