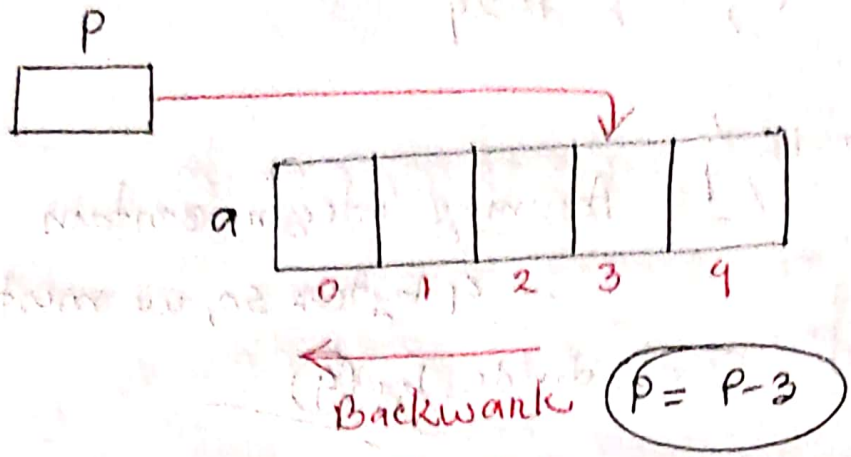
Pointer Arithmetic (Subtraction)



Initially, if p points to a[i], then

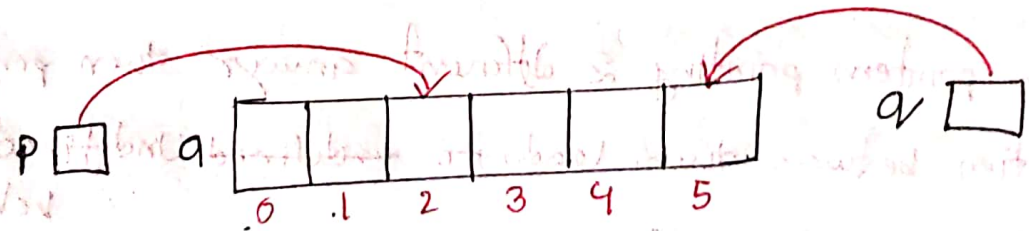
$$p = p - j \equiv \&a[i-j]$$

$$\equiv \&a[3-3] = \&a[0]$$

$$p = 1002 - 3 \times 4 = 1000$$

[Integer contain 4 byte]

⇒ Subtracting one pointer from another,



$q - p = 3$

$p - q = -3$

| just subtraction of index

let P = 1008, Q = 1024

Q - P = 16/4 Assuming integers contain 4 bytes so, we must divide by (4)

~~Hard~~

⇒ Performing arithmatic on pointers which are not pointing to array element leads to undefined behaviour

```
int main() {
    int i = 10; // i is a variable // not array
    int *p = &i;
    printf("%d", *(p+3)); // Not any so (p+3) doesn't exist
    return 0;
}
```

This will occur undefined behaviour.

⇒ If two pointers pointing 2 different arrays then performing subtraction between them leads to ~~undefined~~ Undefined behaviour.

```
int a[7] = {1, 2, 3};
int b[] = {6, 7, 8};
int *p = &a[0];
int *q = &b[0];
```

```
printf("%d", q-p);
```

Undefined behaviour