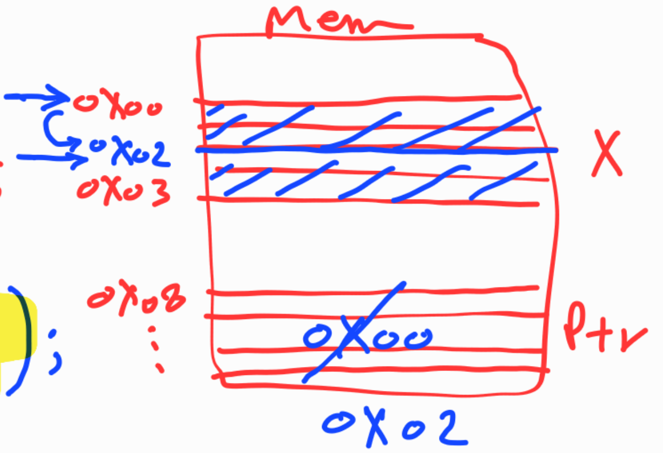
"Pointers - P2"

⇒ Subscriptor ope - [ ]

ex

```
int x = 257;
short *Ptr = &x;
printf("%i", *(Ptr+1));
```

└→ 0 ✓



00000001
00000001
00000000
00000000

Ptr[i]

Ptr[2] = *(Ptr + 2)

$$\boxed{Ptr[i] = *(Ptr+i)}$$

"Pointer Types"

[1] Null Pointer

ASCII of '\0' ← '\0' ← Null char

└→ = Zero = $0000\ 0000)_2$

printf("%c", '\0'); → [illegible]

0 ← Ptr

∴ **Null Ptr**

```
int *Ptr = 0; ≡ int *Ptr = NULL;

Printf ("%d", Ptr); ——→ zero

________________ *Ptr);
                   ↓
                  0x00
           ↓                ↓
        windows            Mac
           ↓                ↓
         crash         exept Int_
```

(Run time error)

AdV

↳ Init_ Pointer ——→ Null

```
int * Ptr = NULL;
  ↳ check < NULL → error
          < != Null → ✓
```

2 wild Pointer

↳ any Pointer that is used before init_

ex

```
void main() {
    int * Ptr;          ← Local → stack → garbage
    Printf ( "%P", Ptr);   → garbage
    *Ptr = 1;  ←
}
```

#

hint

* operators with Pointer

① ( )
② ++ , --
③ * (dereference)

① *ptr++ → ++ > *

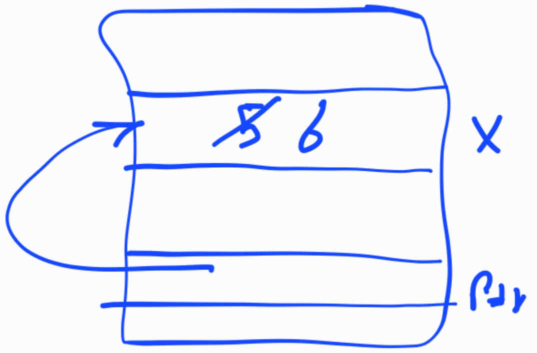
but will be implement in the next line

② *++ptr → ~ ~ ~

& will be implement in the same line

③ (*ptr)++

→ ( ) is higher Pern than (++)



④ *ptr1 ± *ptr2

→ (*) is ~ ~ (++)

* Const with Pointer

Value is Const

- int const *ptr = &x;
- const int *ptr = &x;

Pointer is Const

- int *const ptr = &x;

```
Const int X = 10;
int * Ptr = &X ;
*Ptr = ✓ ;   || X = ✓
```

```
int x = 5;
Const int *Ptr = &X;
*(Ptr) = 7 ;   XX
*(int *) Ptr = 7;   ✓✓
```

⊛ Pointers with funcs

↳ Arg ⇒ Pass by ref

ANSI-C

↳ return ⇒ only one return

Sol ⟶ Pointer as an Arg

func return Addr

في func ⟹

```
int * func (void)
{
static int x;
    return &x;
}
```

```
void main()
{
    int * ptr = func();
    printf("%d", *ptr);
}
```

3 dangling Pointer

→ Pointer → Points to a deleted / de-allocated object

Sol :-

→ local → (+ static)