CS & IT ENGINEERING

Programming in C

String in C Programming

Lec- 01

By- Pankaj Sharma sir

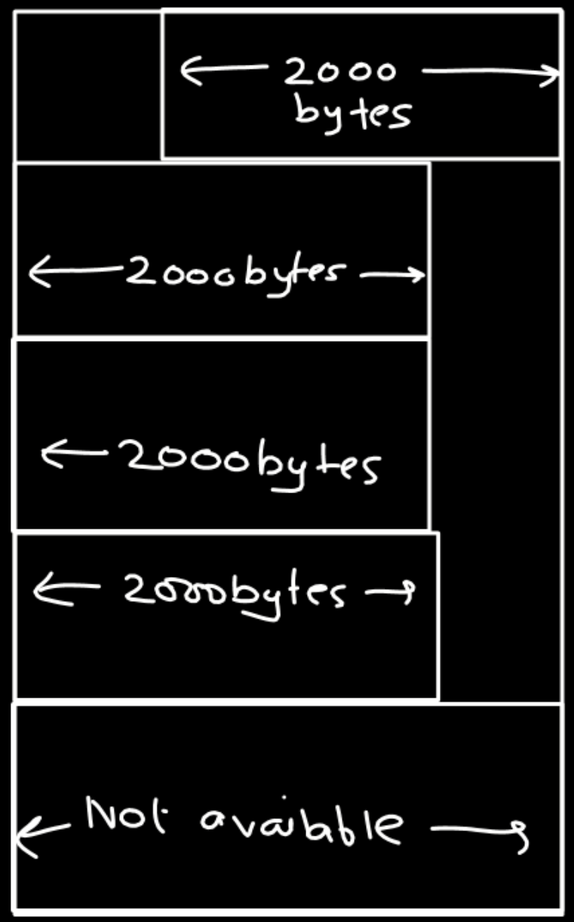
Strings-1

free

```
void fun() {
    int *p;
    P = malloc(500 x sizeof(int));
    ||
    ||
    ||
    }
```

Memory leakage problem

P = local variable



Heap

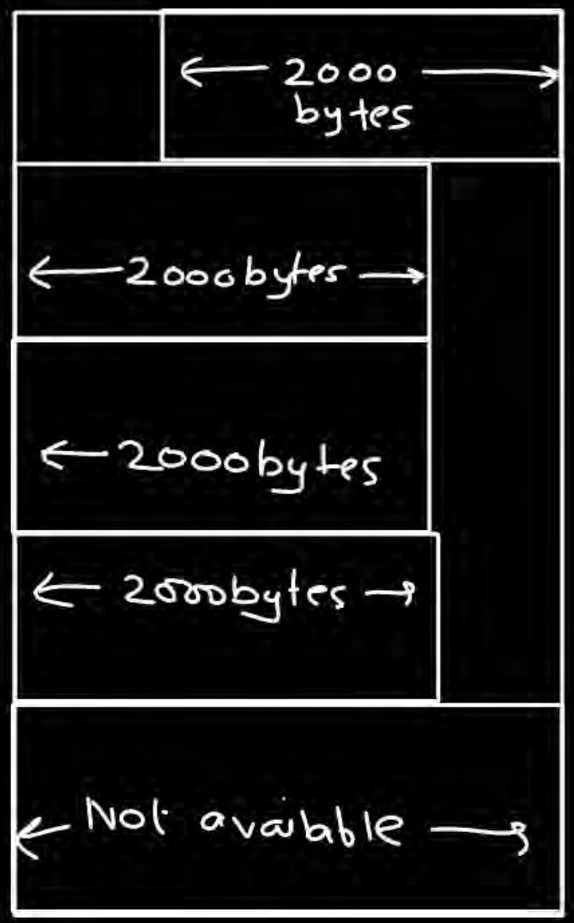
```
Void main()
{
① fun();
② fun();
③ fun();
  fun();
  fun();
}
```

free

```
void fun() {
    int *p;
    p = malloc(500 * sizeof(int));
    ||
    ||
    ||
    free(p);
}
```

p = local variable

Memory leakage problem



Heap

```
Void main()
{
① fun();
② fun();
③ fun();
  fun();
  fun();
}
```

Strings

* Sequence of characters terminated by NULL character.

Collection/group of characters

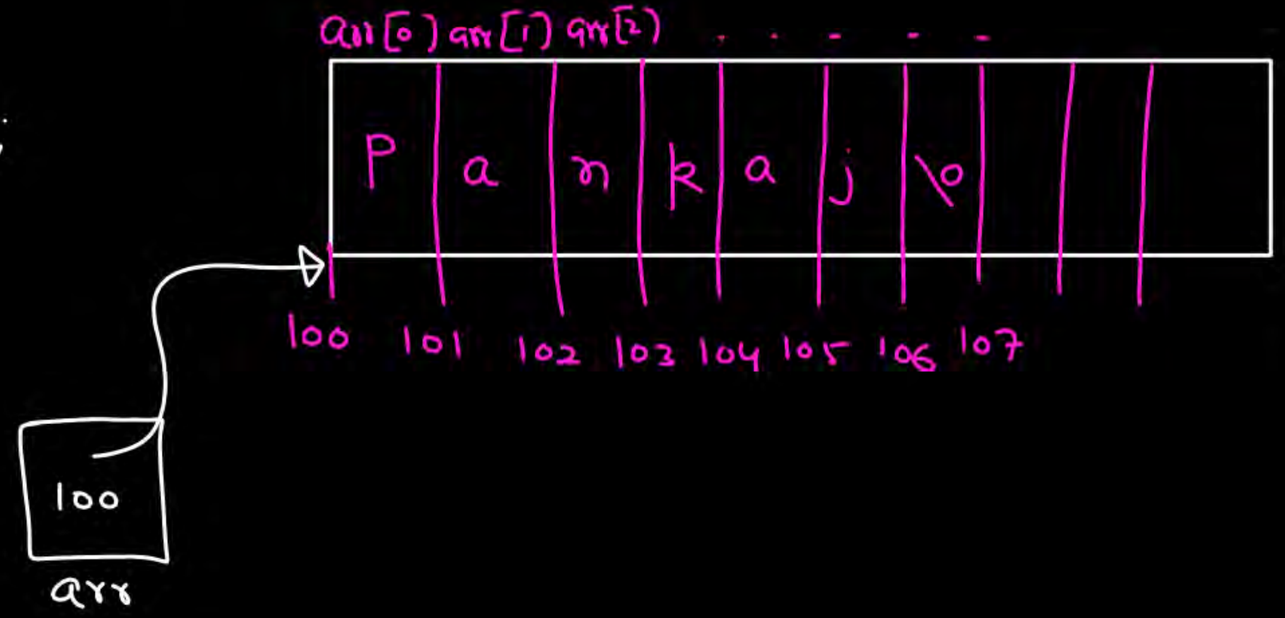
'\0' ⇒ Ascii code 0.

(1)

```
char arr[10] = "Pankaj";
```

string literal

```
printf("%s", arr);
```



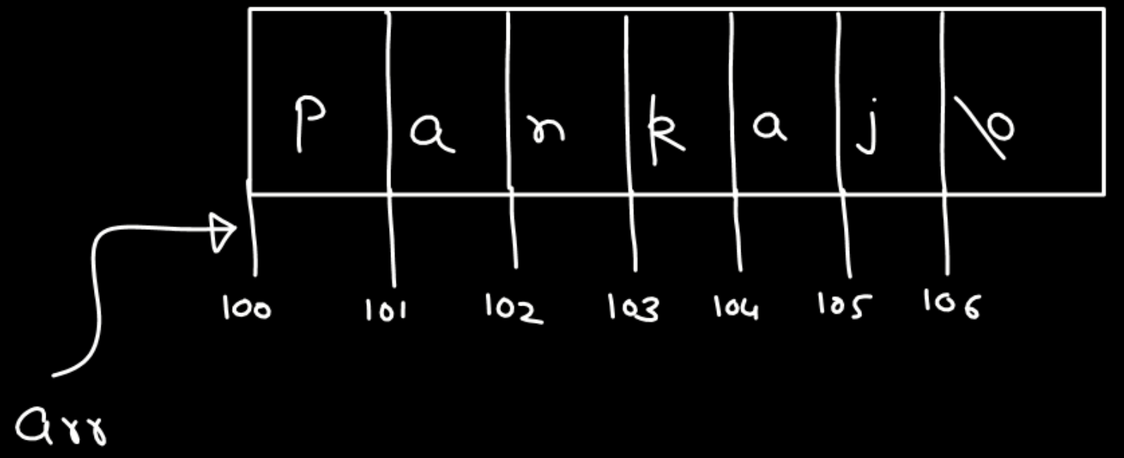
1

```
char arr[ ] = "Pankaj" ;

printf("%s", arr);
```

string literal

6+1 → '\0'



(1)

```
char arr[ ] = "Pankaj" ;
```

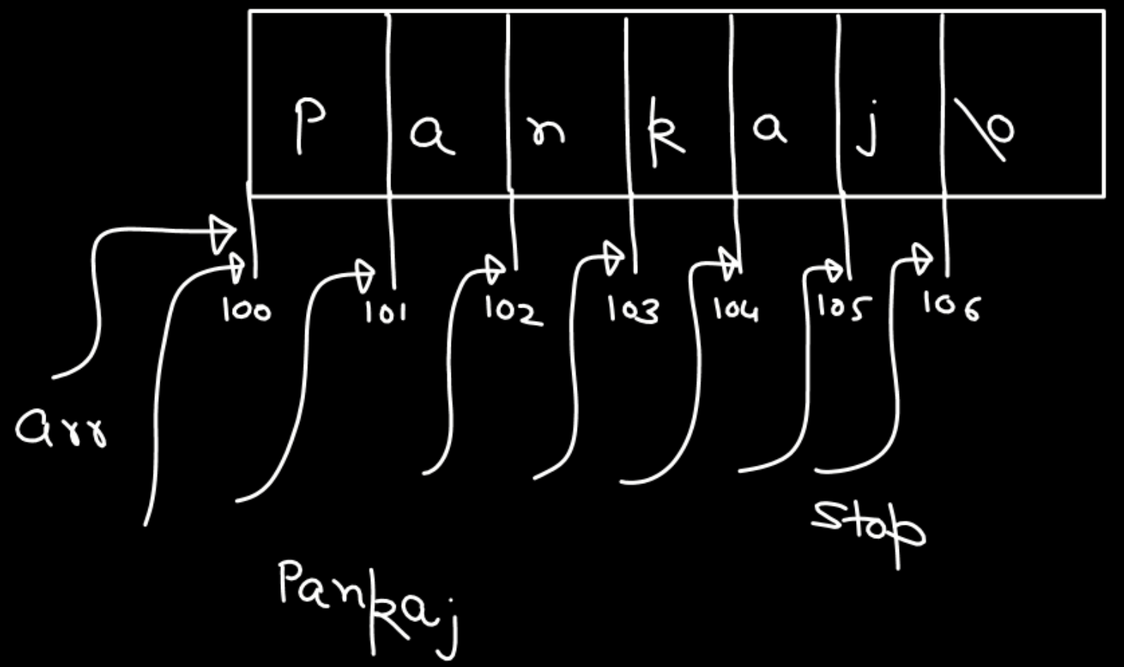
string literal

6+1 → '\0'

```
printf("%s", arr);
```

&arr[0]

Address



(1)

```
char arr[ ] = "Pankaj" ;
```

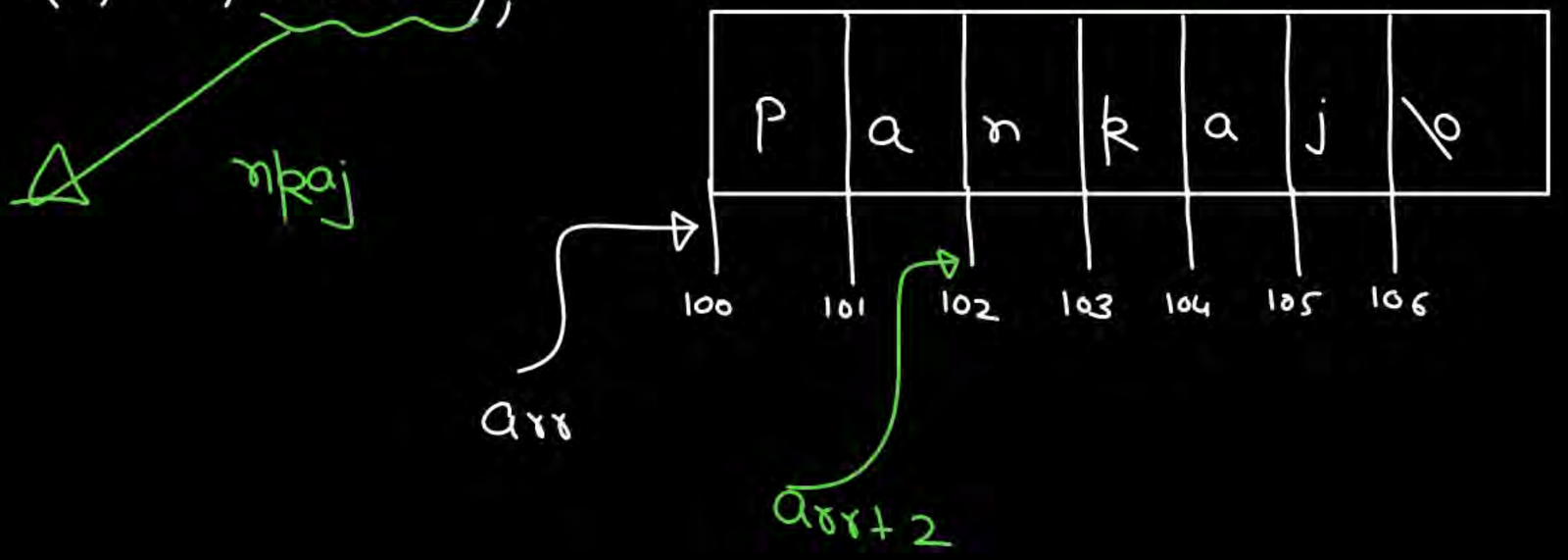
string literal

6+1 character

'\0'

```
printf("%s", arr+2);
```

nkaj



String literal

1. `char arr[20] = "Pankaj";`
2. `char arr[20] = "Pankaj" "sharma" "sir";`
3. `char arr[] = "Pankaj \`
   `sharma \`
   `sir";`

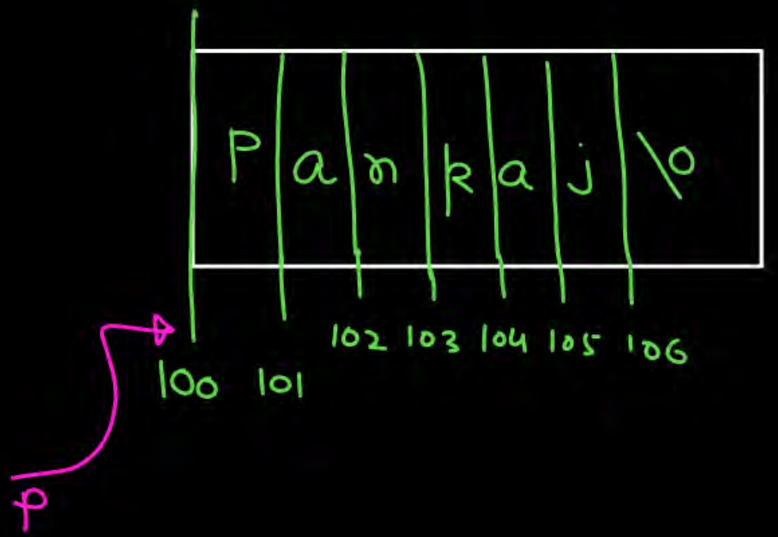
(2)

Address of 'p'

```
char *p = "Pankaj";
```

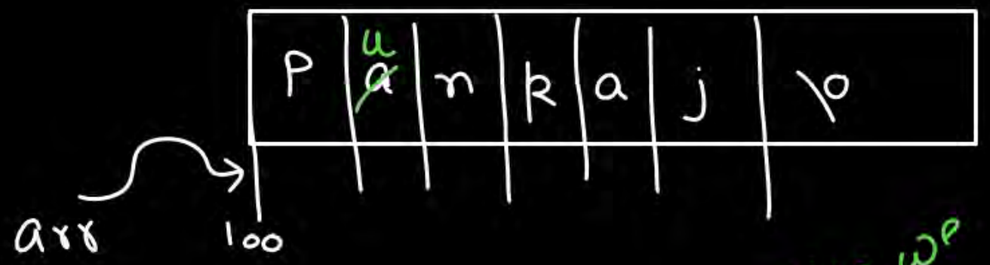
```
pf("%s", p)
```

Pankaj



char arr[] = "Pankaj";

Read/write



arr++ ; Invalid

++arr ; X

--arr ; X

arr-- ; X

Can we update individual element of array ✓

arr[1] = 'u';

arr = "Neeraj"; Invalid

Lvalue ✓

int a[] = {10,20,30};

a[0] = 1000; ✓

a[1] = 2000; ✓

a = {40,50,60}; Invalid

Lvalue ✓

```
char arr[] = "Pankaj";
```

① ind. elements can be updated

```
arr[1] = 'u';
```

✓

② Completely new string can't be assigned to array.

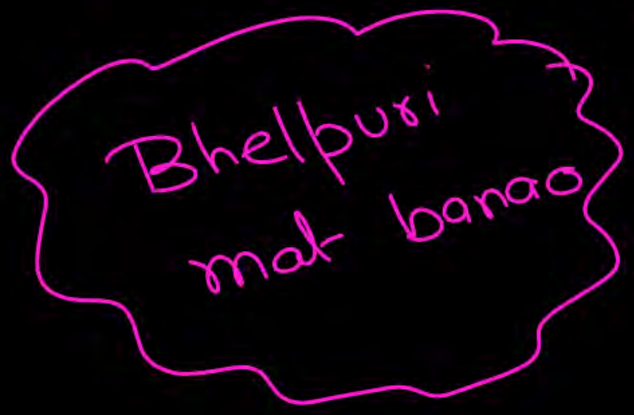
```
arr = "Neeraj";
```

✗

③ Read/Write

④ To avoid any problem: size of array must be atleast 1 size more than the no. of symbols in string.

```
char arr[6] = "Pankaj";
```



1.

```
int arr[] = {1,2};
```

```
char arr[] = {'P','a','n','k','a','j'};
```

6

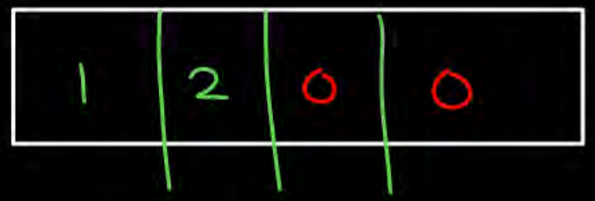

```
bf("%s",arr);
```

→ Pankaj

→ No...

Pankaj□□_

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




```
char arr[7] = {'p','a','n','k','a','j'};
```

0 → int

Symbol

Ascii ⇒ '\0'

Code

Zero ≠

char arr[] = "Pankaj";

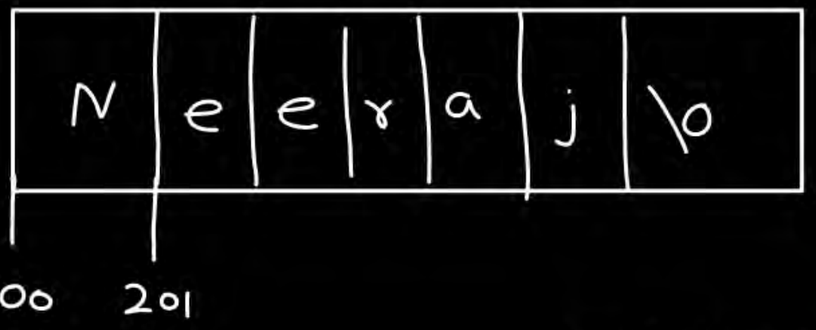
printf("%s", arr); → Address

printf("%s", arr+1); → Address

printf(arr);

printf(arr+1);

char *Ptr = "Neeraj"; (200)



printf("%s", Ptr); → Address

pf("%s", Ptr+1); → Address

pf(Ptr);

pf(Ptr+1);

16 years → 40 hrs

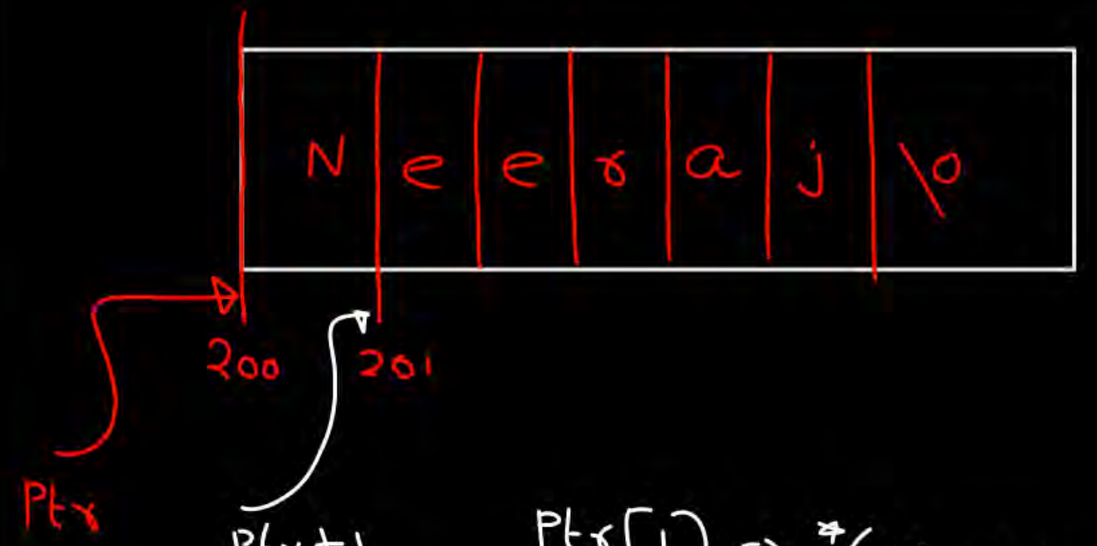
Gate → 100%

char arr[] = "Pankaj"; Read/write

arr[1] = 'u'; ✓

arr = "Neeraj"; Invalid (Error)

char *Ptr = "Neeraj"; Read only area



Ptr[1] ⇒ *(Ptr+1)

Ptr[1] = 'i'; Invalid

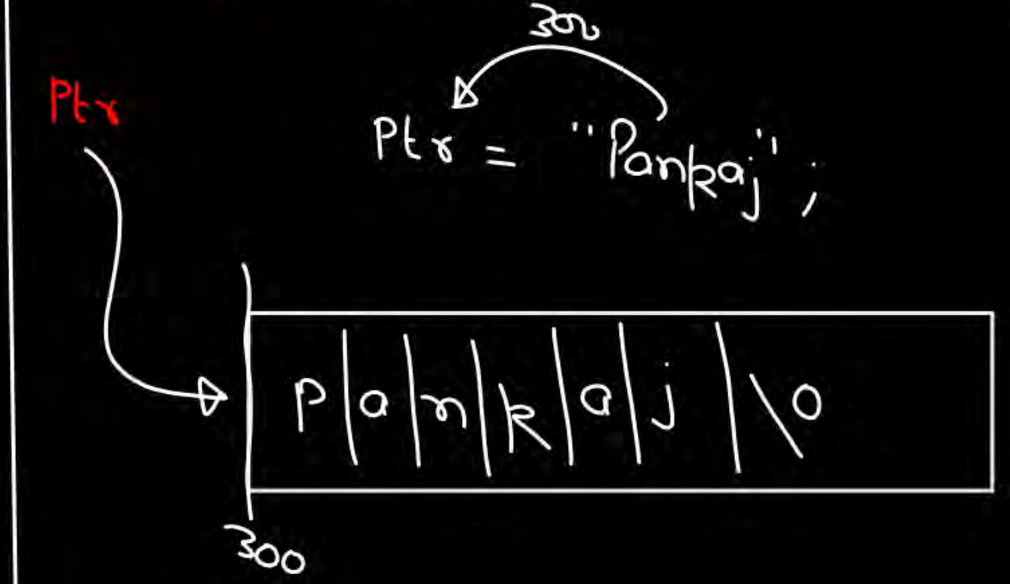
```
char arr[] = "Pankaj";
```
Read/write

```
arr[1] = 'u';
```
✓

```
arr = "Neeraj";
```
Invalid (Error)

```
char *Ptr = "Neeraj";
```
Read only area

Ptr ⇒ Pointer variable



```
void main(){
    printf("Hello");
    printf("Hello"+1);
}
```

literal → Read only Area

printf("Hello"+1); → ello

"Hello" → Address of 'H'

"Hello" + 1 → Address of 'e'

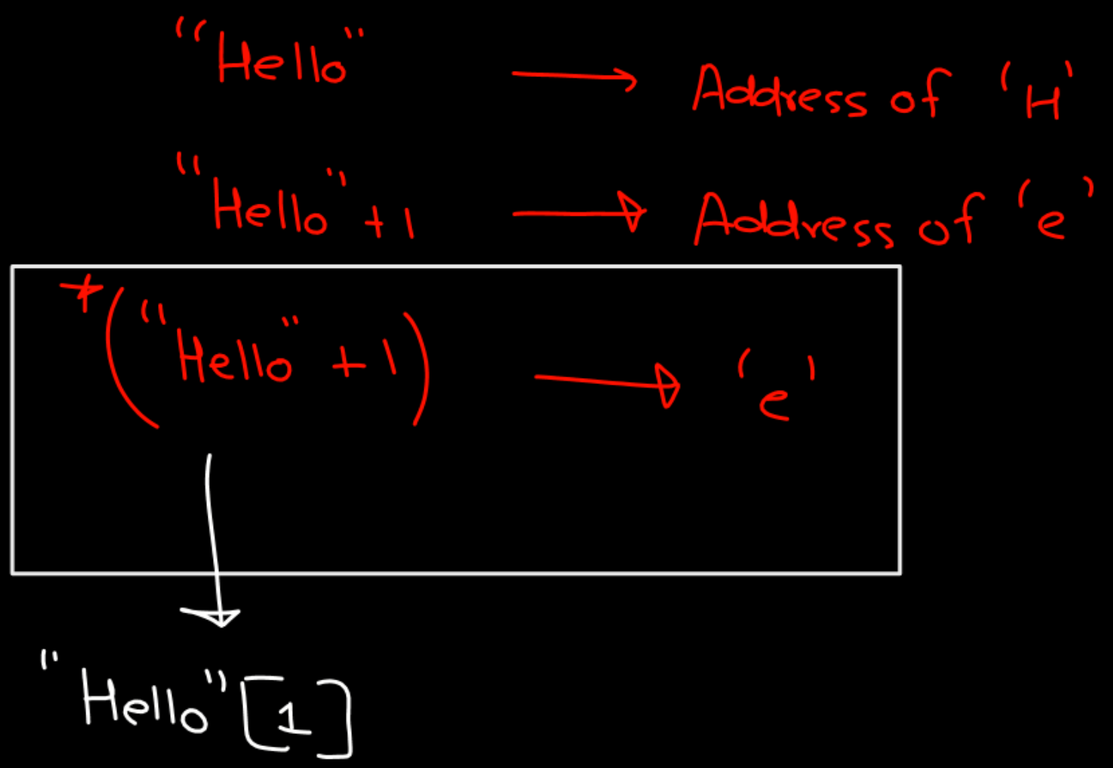
P[i] => *(P+i) => value at ——

printf("%c", *("Hello"+1));

—

*(x+i)

= x[i]




THANK YOU GW
SOLDIERS !