CS & IT ENGINEERING

Programming in C

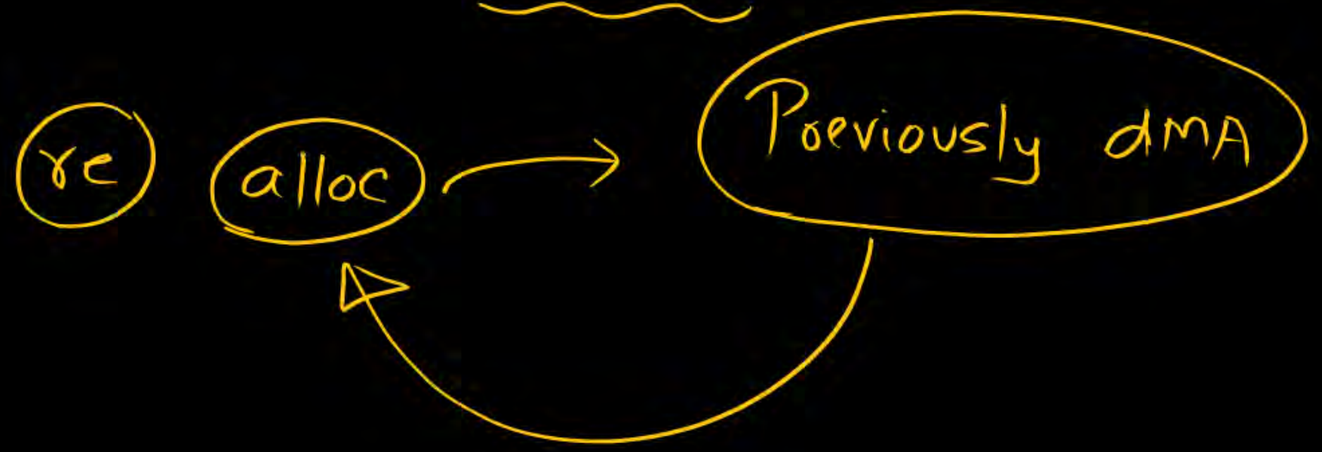
Strings

Lec- 01

By- Pankaj Sharma sir

- Strings-1

realloc



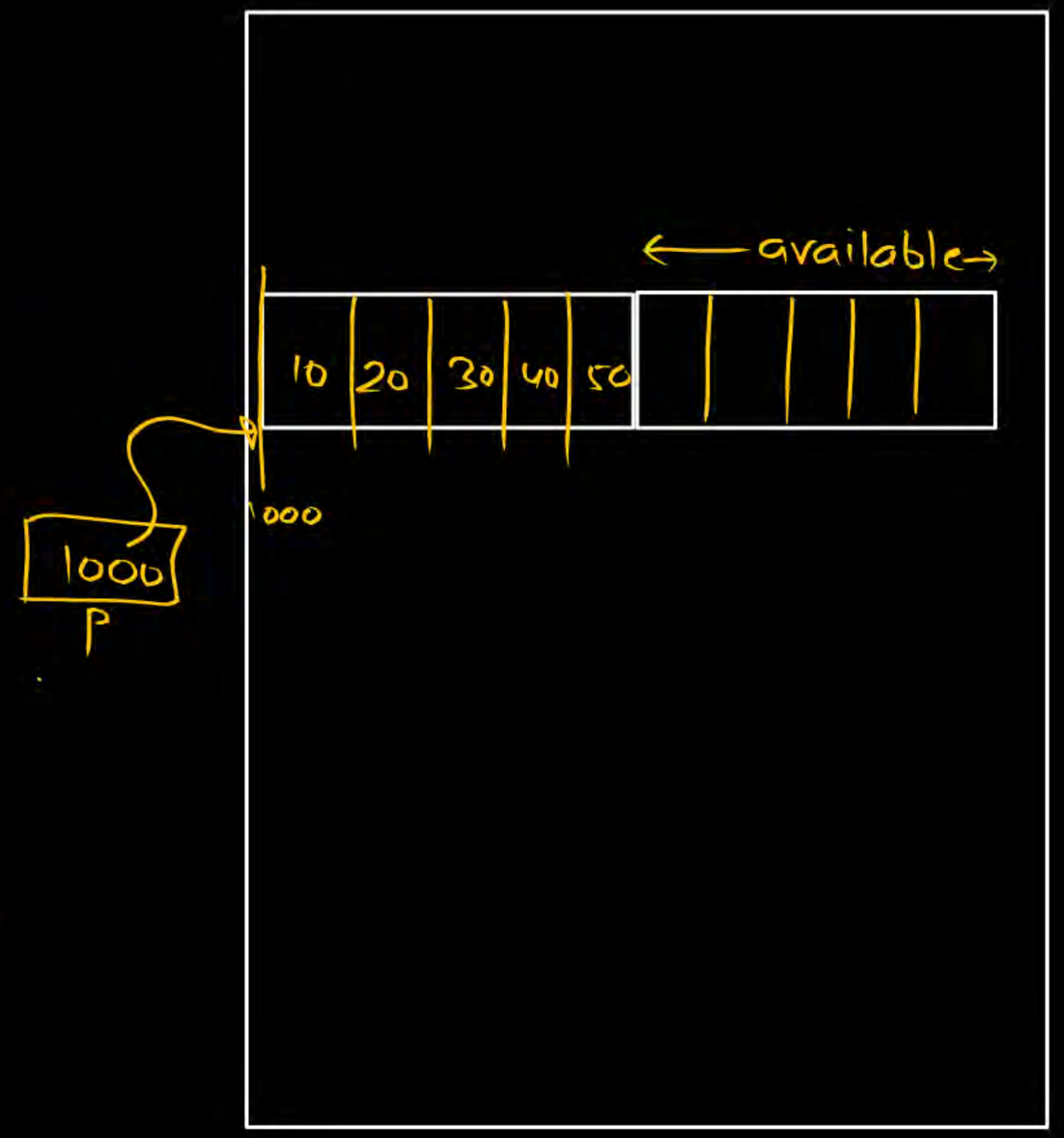
```
int *p;
P = malloc( 5 x sizeof(int));
```

1000

```
P = realloc( P, newsize)
P = realloc (P, 10x sizeof(int));
```

case i)

1000

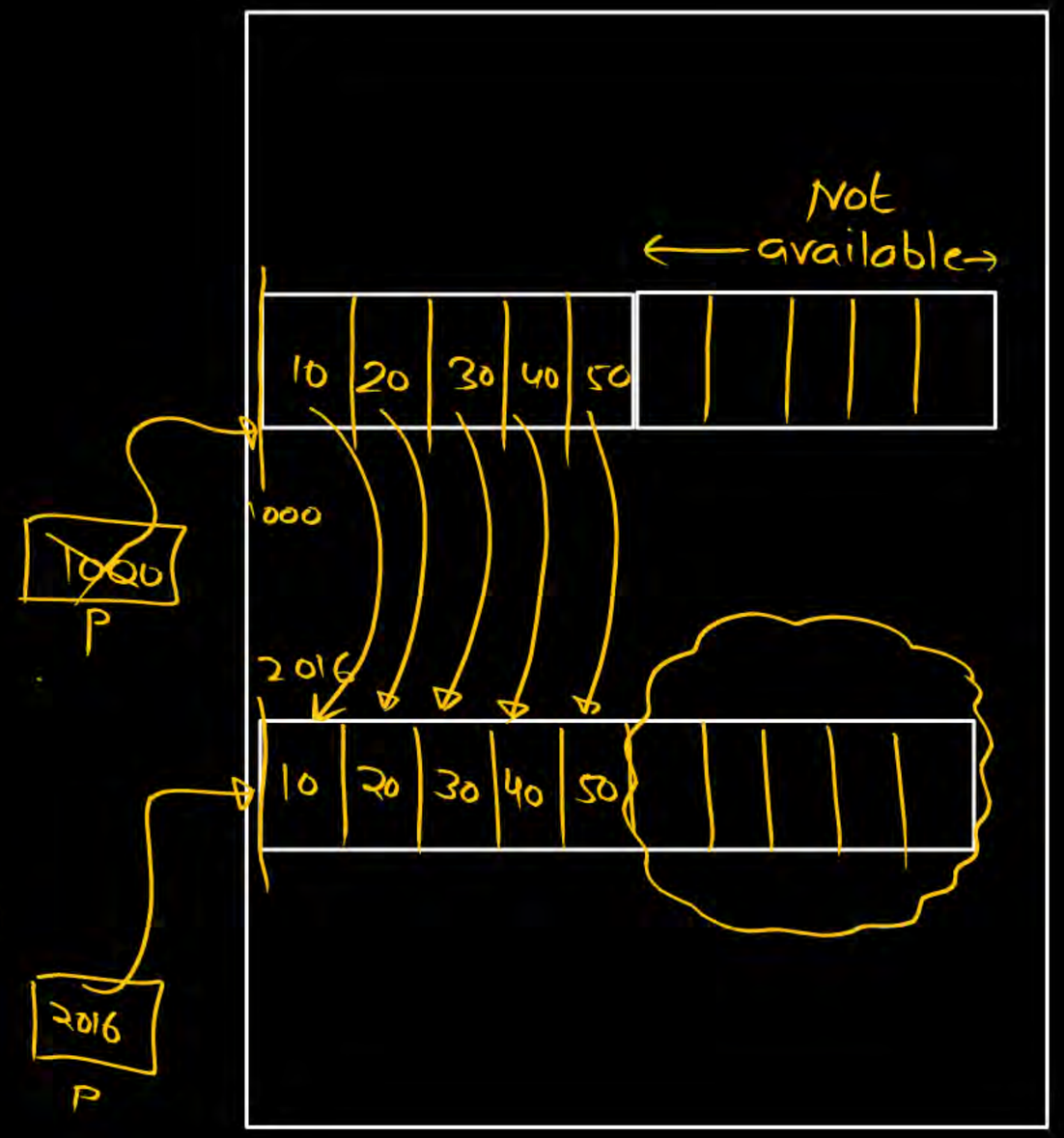


```
int *p;
P = malloc( 5 x sizeof(int));
```

1000

```
P = realloc( P, newsize)
P = realloc (P, 10x sizeof(int));
```

Case 2)

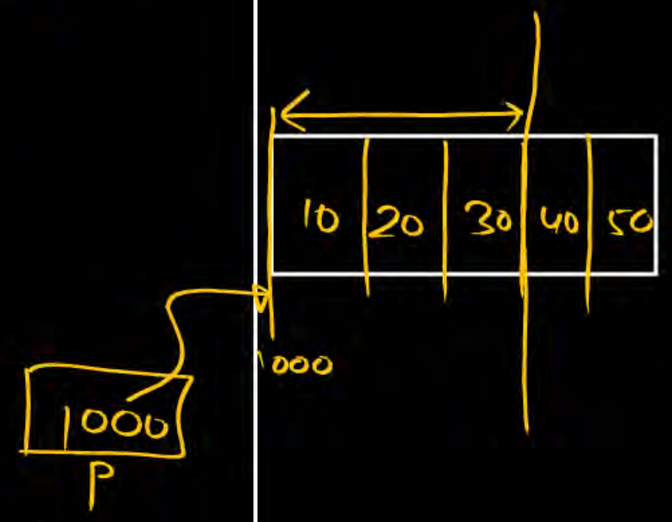


AIR-1

```
int *p;
P = malloc( 5 x sizeof(int));
```

1000

```
P = realloc( P  newsize)
P = realloc(P, 3 x sizeof(int));
```



```
void f(){
    int *p;
    p = malloc(1000x sizeof(int));
    ≡
}
void main(){
    f();
    f();
    f();
    f();
}
```

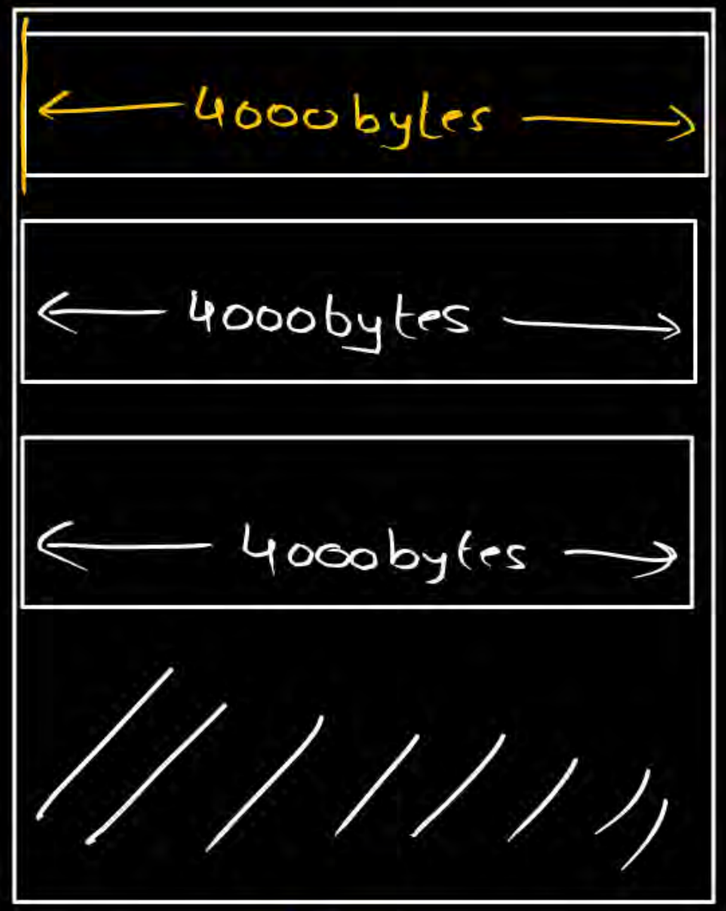
local var.

Heap → throughout the program

Memory leakage Problem

Memory not available

free



Heap

```
void f(){
    int *p;
    p = malloc(1000x sizeof(int));
    ≡
    free(p);
}
void main(){
    f();
    f();
    f();
    f();
}
```

local var.

Strings

A seq. of characters terminated by null character.

→ \0

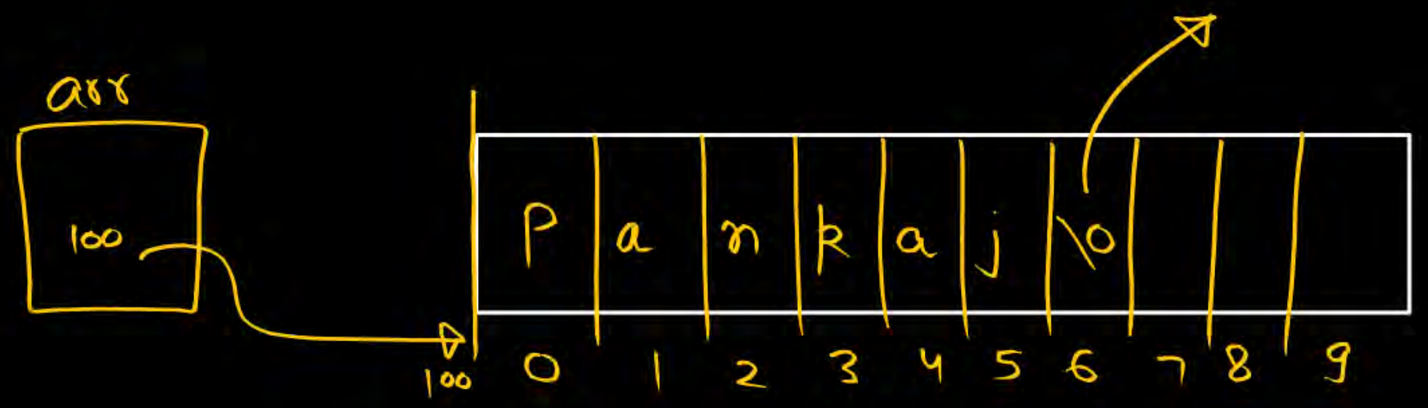
Ascii code ⇒ 0

Collection/group of characters

a	b	c	\0

string constant

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




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

Pankaj

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

P	a	n	k	a	j	\0

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

Atleast 1 more than string size

behaviour is undefined

```
char arr[ ] = {'p', 'a', 'n', 'k', 'a', 'j'};
printf("%s", arr);
```

```
char arr[ ] = {'p', 'a', 'n', 'k', 'a', 'j', '\0'};
printf("%s", arr);
```

① char name[] = "Pankaj";

pf("%s", name);

→ address

② char name[] = {'P', 'a', 'n', 'k', 'a', 'j', '\0'};

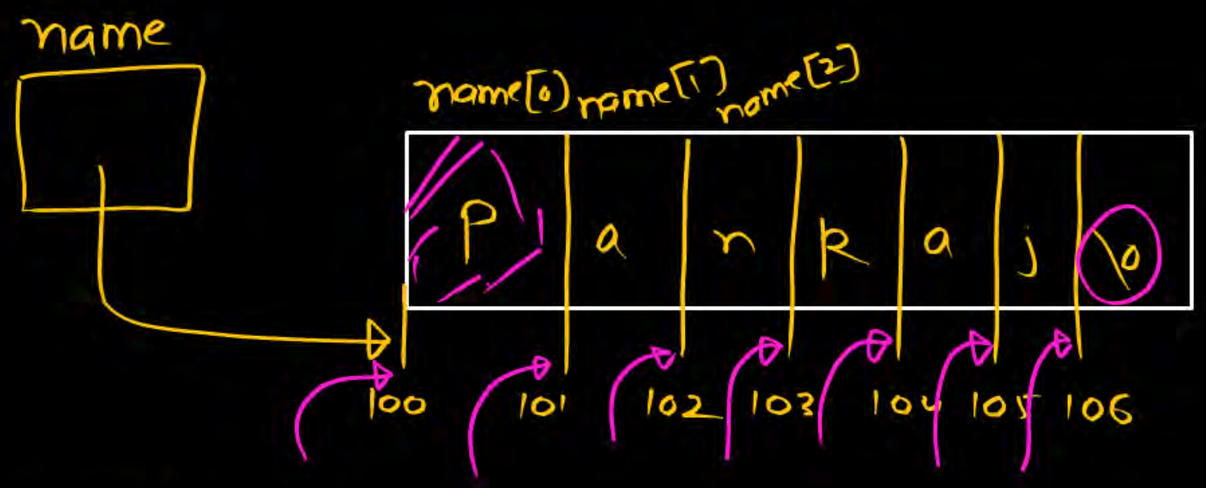
pf("%s", name);

→

③ char name[10] = "Pankaj";

pf("%s", name);

→



Pankaj

for string

pf को address देता है

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

pf("%s", name+1);
```

$\Rightarrow$ ankaj

P	a	n	k	a	j	\0
100	101	102	103	104	105	106

name = &name[0]

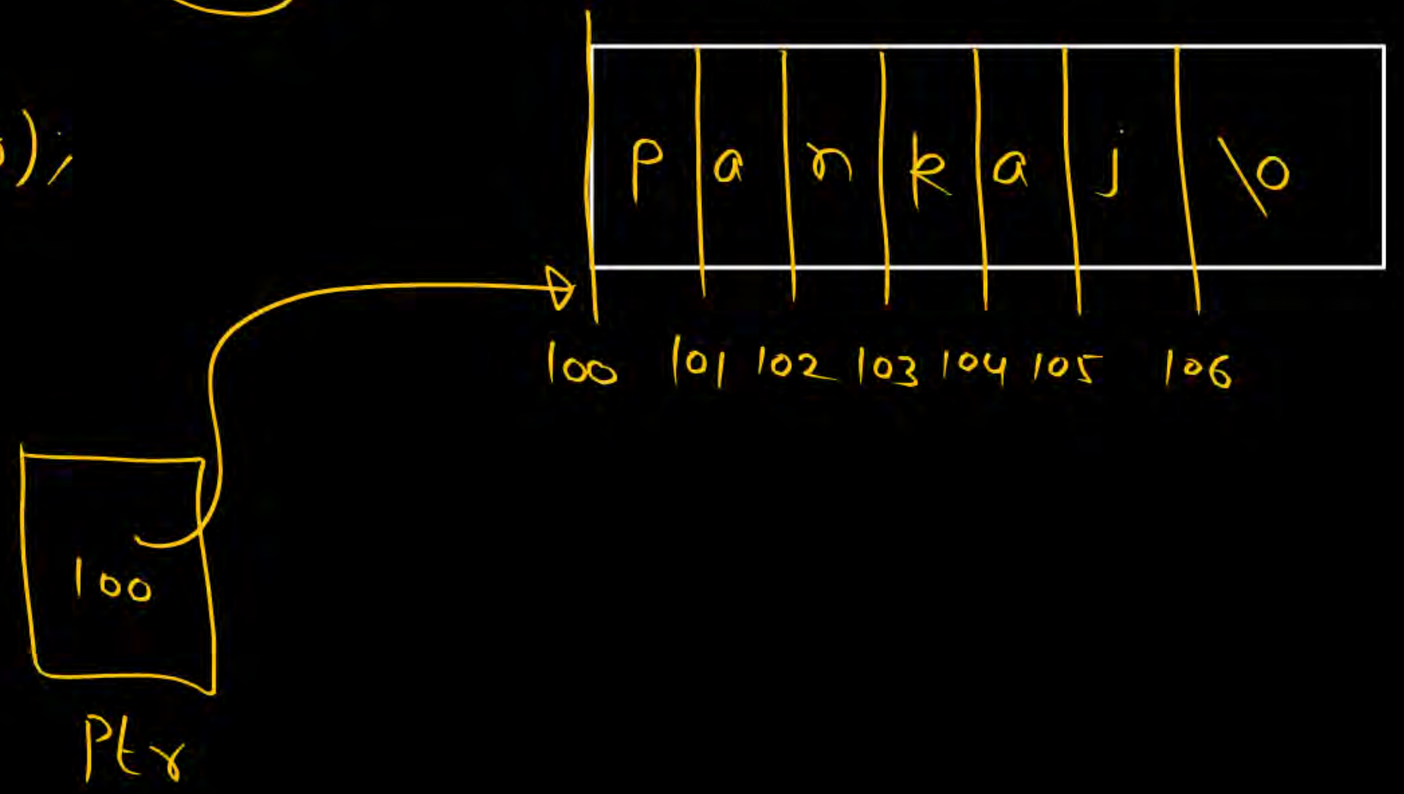
name+1 $\Rightarrow$ &name[0]+1

= &name[1]

```
char *ptr = "Pankaj";
```

100

```
pf("%s", ptr);
```



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

```
pf("%s", arr);      Neeraj
pf("%s", arr+1);    eeraj

pf(arr);
pf(arr+1)
```

```
char *ptr = "Neeraj";
```

```
pf("%s", ptr);      Neeraj
pf("%s", ptr+1);    eeraj

pf(ptr);
pf(ptr+1)
```

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

```
++arr;
arr++;
--arr;
arr--;
```

```
char *Ptr = "Neeraj";
```

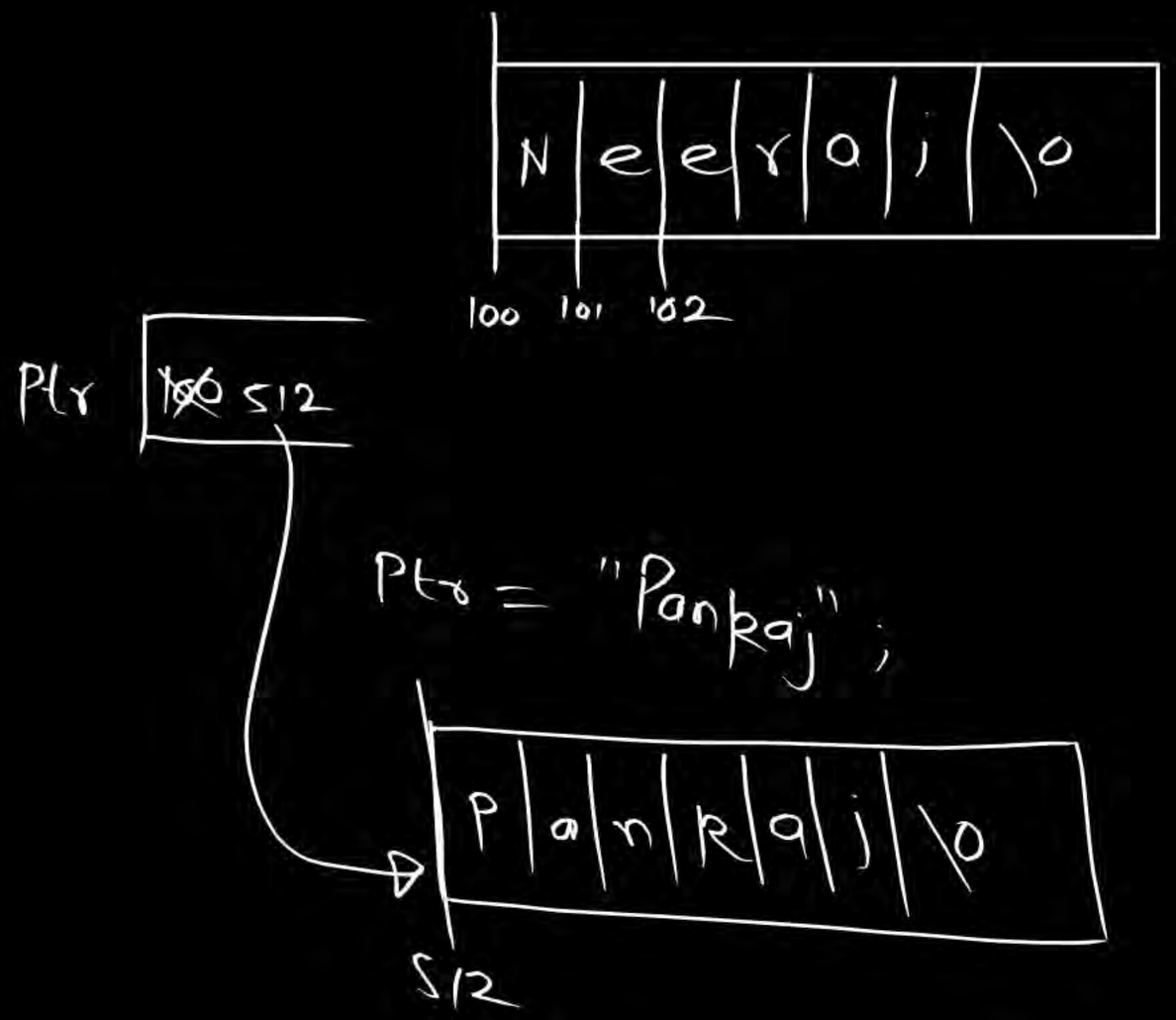
```
Ptr++;
++Ptr;
--Ptr;
Ptr--;
```

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

arr = "Pankaj"; Invalid

L value can not be array-name

```
char *Ptr = "Neeraj";
```



Read/write

```
int arr[4] = {10,20,30,40};
arr[1] = 1056;
```

arr[0]	arr[1]	arr[2]	arr[3]
10	1056 ~~20~~	30	40

Can we change content/ Individual element of an array

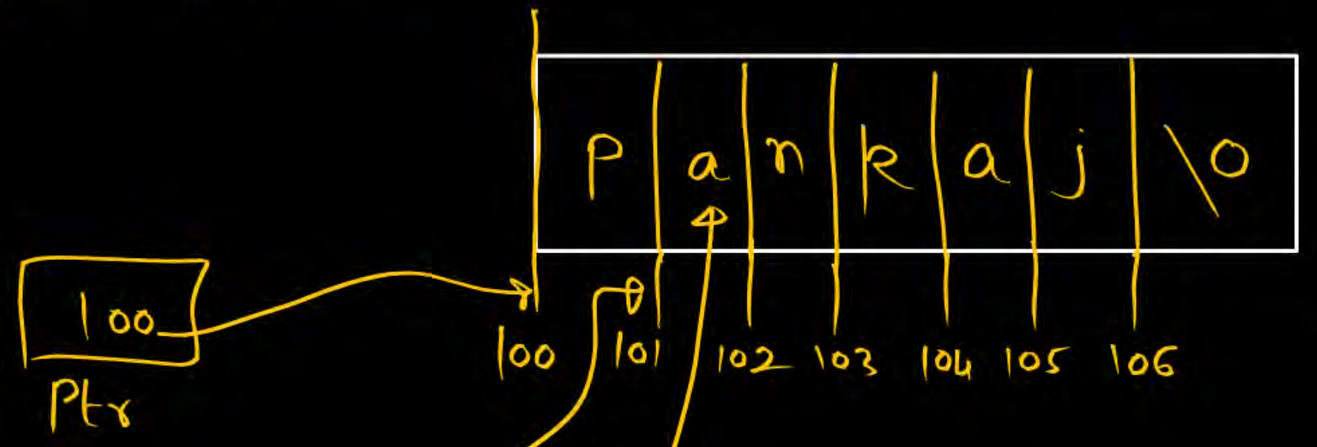
→ YES

arr = {100, 200, 300, 400};

Lvalue ✗

char *Ptr = "Pankaj";

Read only area



Ptr+1

*(Ptr+1)

⇓

Ptr[1] = 'u';

// Don't try this

```
char arr[ ] = "Pankaj";
arr[1] = 'u';
```

array

1. Individual element को change
2. completely new string assign नहीं कर सकते
3. ++array_name
   --array_name
   array_name++
   array_name--
4. Read/write area

Pointer

1. we can't change individual char.
2. we can assign a new string.
3. ++ptr ✓
   --ptr
   ptr--; ✓
   ptr++; ✓
4. Read only area

```
void main(){

    printf("Hello");

    }
```

→ string constant/literal

→ Read only Area

```
void main(){

    printf("Hello");

}
```

H	e	l	l	o	\0
100	101	102	103	104	105

"Hello" → Address of 'H'
i.e 100

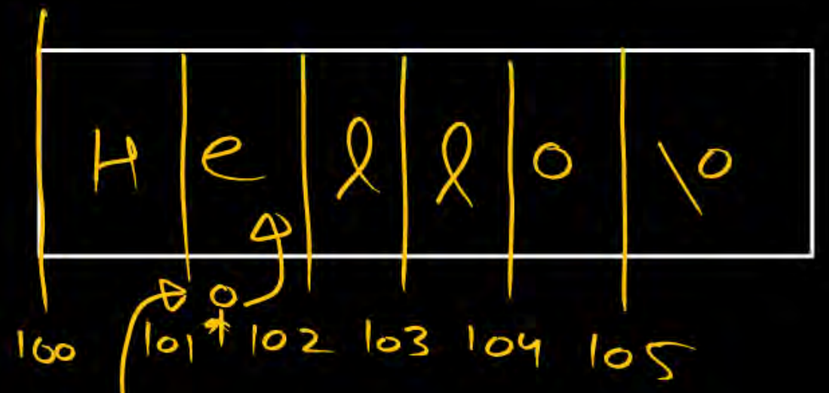
"Hello" + 1 → Address of 'e'
i.e 101

pf("Hello" + 1) ⇒ ello

```
void main(){

    printf("Hello");

}
```



"Hello" → Address of 'H'
i.e 100

"Hello" + 1 → Address of 'e'
i.e 101

*("Hello" + 1) → 'e'

"Hello"[1] ⇒ 'e'

*(a+i)
⇒ a[i]

pf("%c", "Hello"[1]);
o/p : e

'n' == 'n'

```
if ("Pankaj"[2] == "neeraj"[0])
        printf("Pankaj Sir");
else
        printf("Gate Walloh");
```

Kunal

%c → symbol
Char
System

%d → int

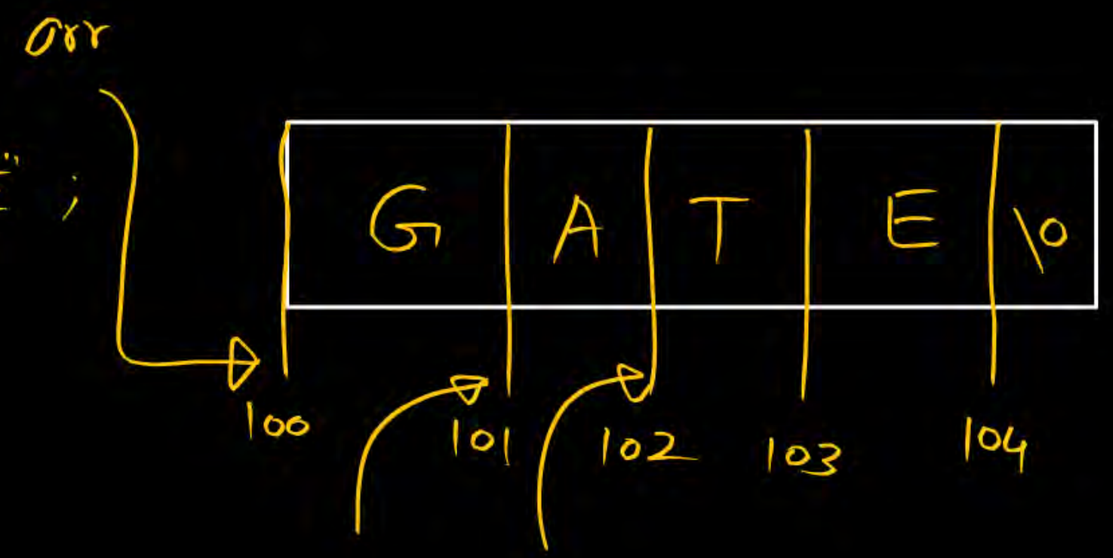
Q

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

printf("%s", arr);
printf(arr);
printf(arr + 1);
printf(arr + 3-1);
```

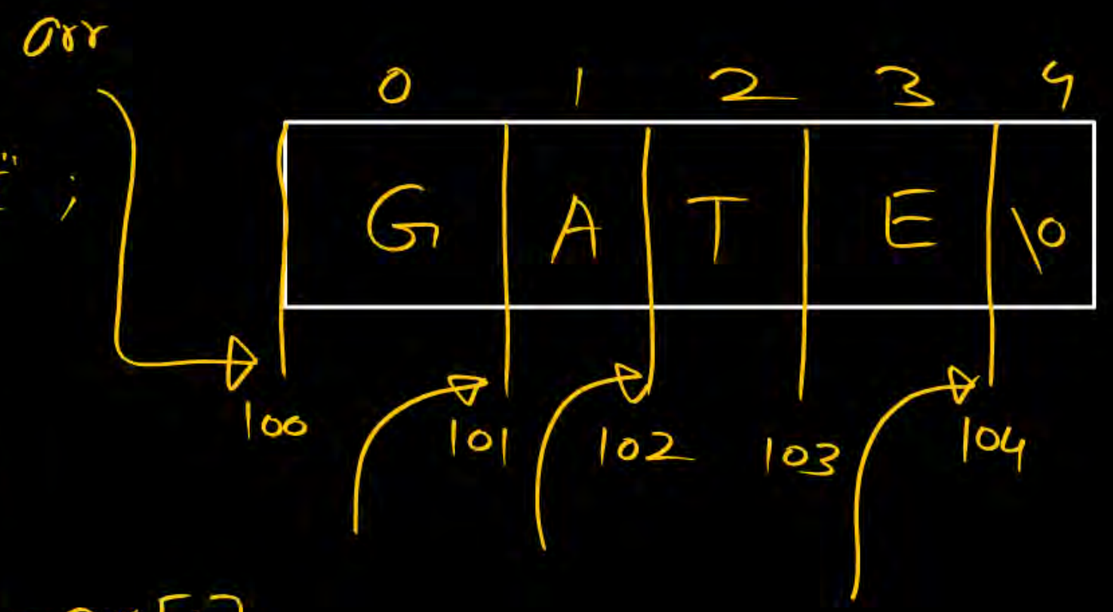
TE

arr + 2



Q

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




```
pf(arr + 3[arr] - arr[1])
```

arr + 3[arr] - arr[1]

No o/p

arr + 'E' - 'A'

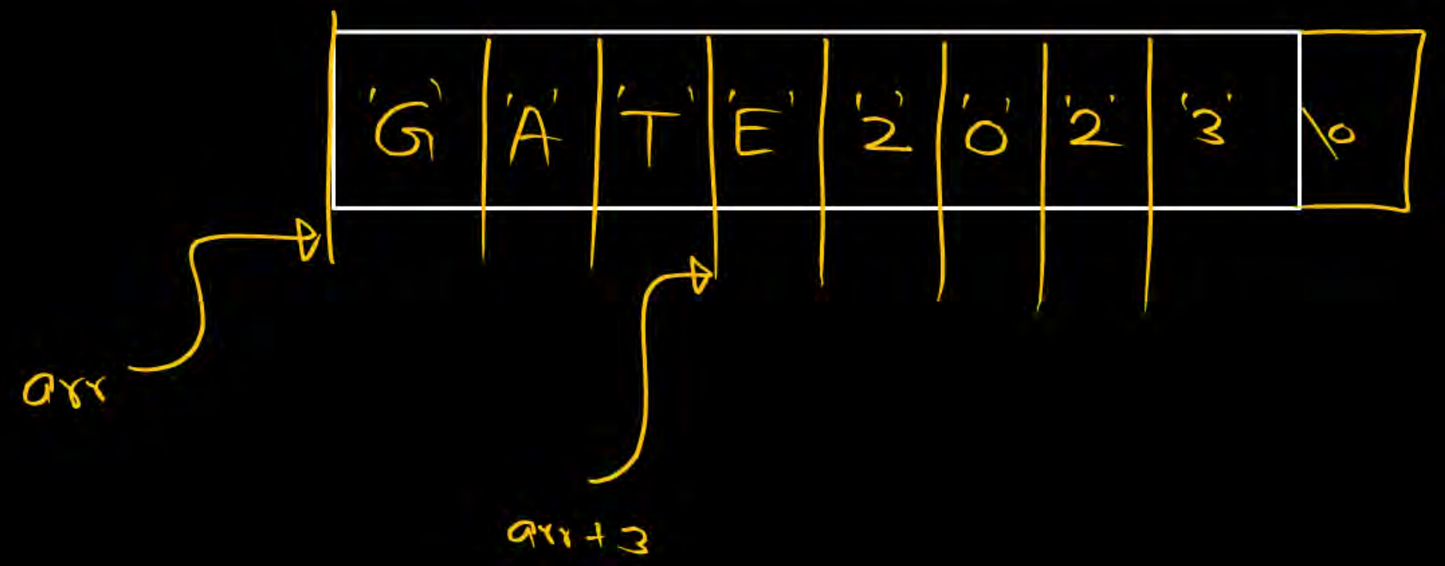
arr + x+4 - x

⇒ arr+4

A - x
- x+1
- x+2
- x+3
E - x+4

Q

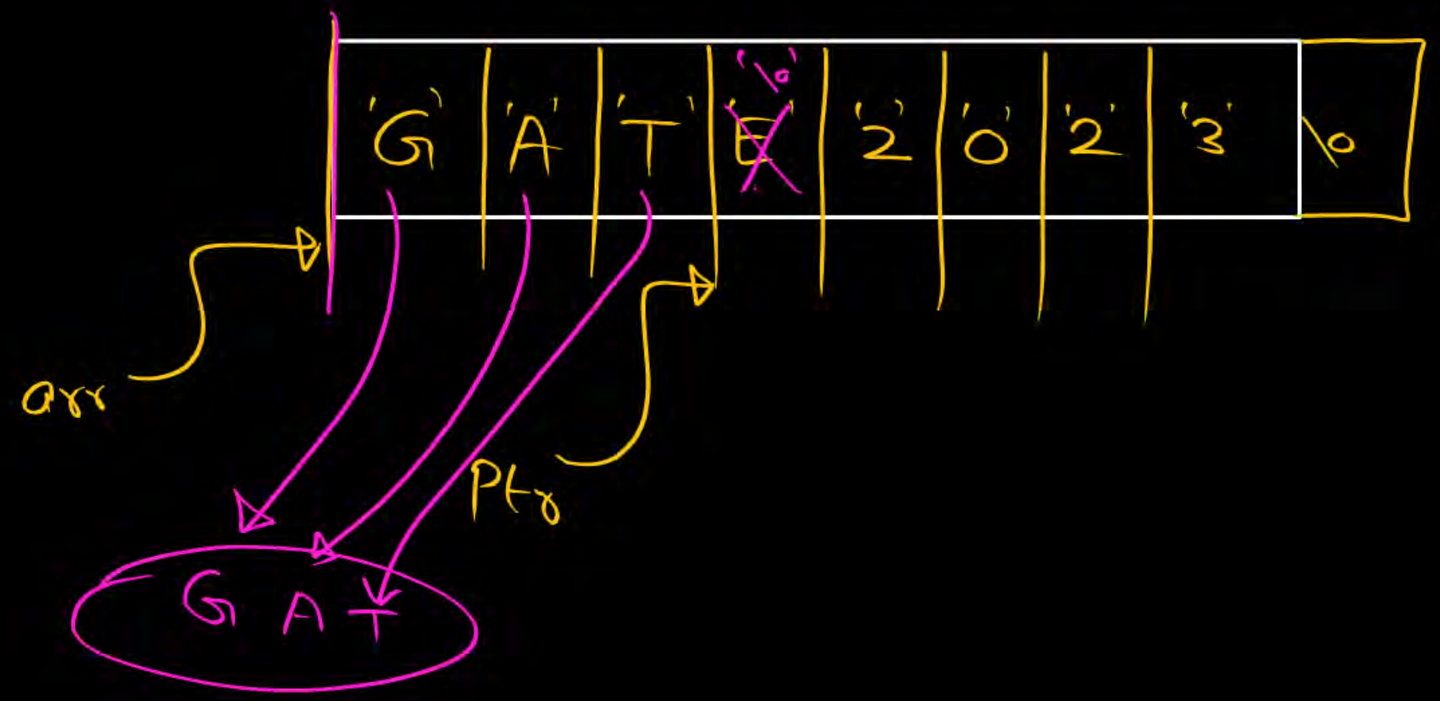
```
char arr[] = "GATE2023";
char *ptr;
ptr = arr + 3;
*ptr = '\0';
pf(arr);
```



Q

```
char arr[] = "GATE2023";
char *ptr;
ptr = arr + 3;
*ptr = '\0';
bf(arr);
```

GAT



08 : 30 AM

Strings - 2

9:00 PM

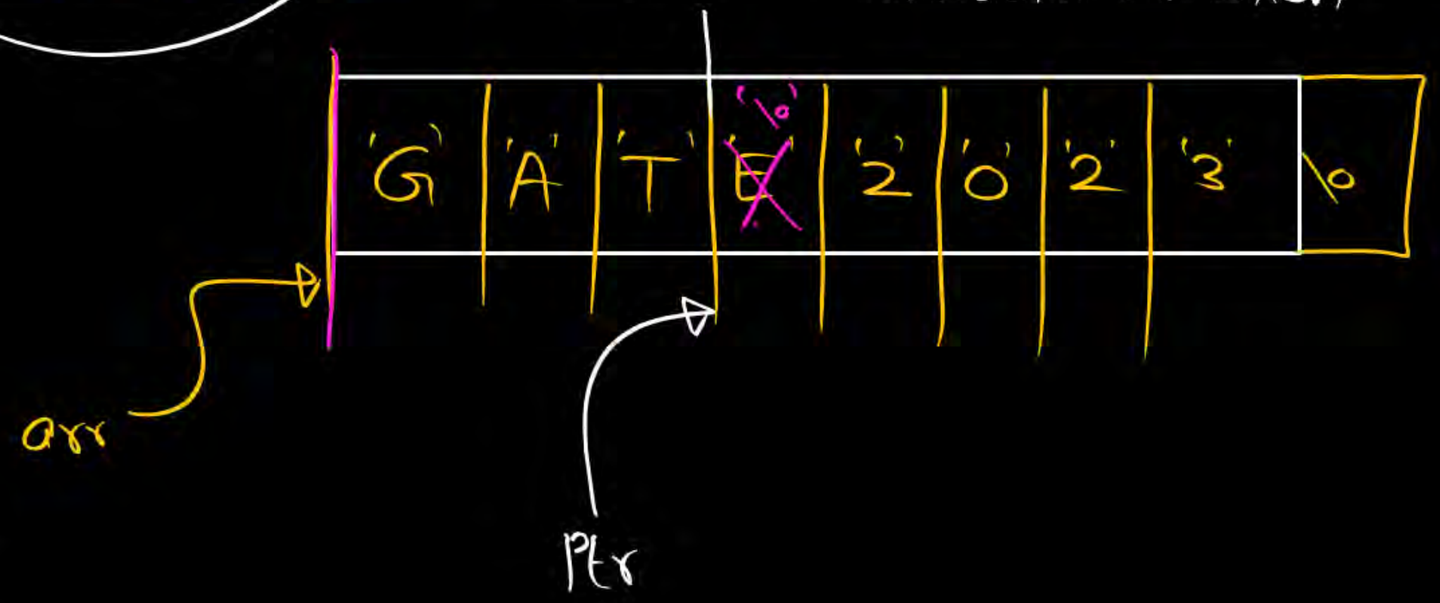
structure & union

Q

```
char arr[] = "GATE2023";
char *ptr;
ptr = arr + 3;
*ptr = '\0';
pf(arr);
pf(ptr);
```

GAT

No o/p



char *ptr


THANK YOU GW
SOLDIERS !