

# CS & IT ENGINEERING

Programming in C

**Arrays and Pointers**

**Lec- 08**



By- Pankaj Sharma sir





TOPICS TO BE  
COVERED



**Arrays and Pointers**

# Dynamic Memory Allocation

int a[5000];

int a[400];

① malloc

② calloc

③ realloc

④ free

## malloc

malloc(size in bytes)  
↗ -ve ↘

malloc(10);

Syntax:

```
void* malloc(unsigned int)
```

Heap



~~`int* p = malloc(10);`~~

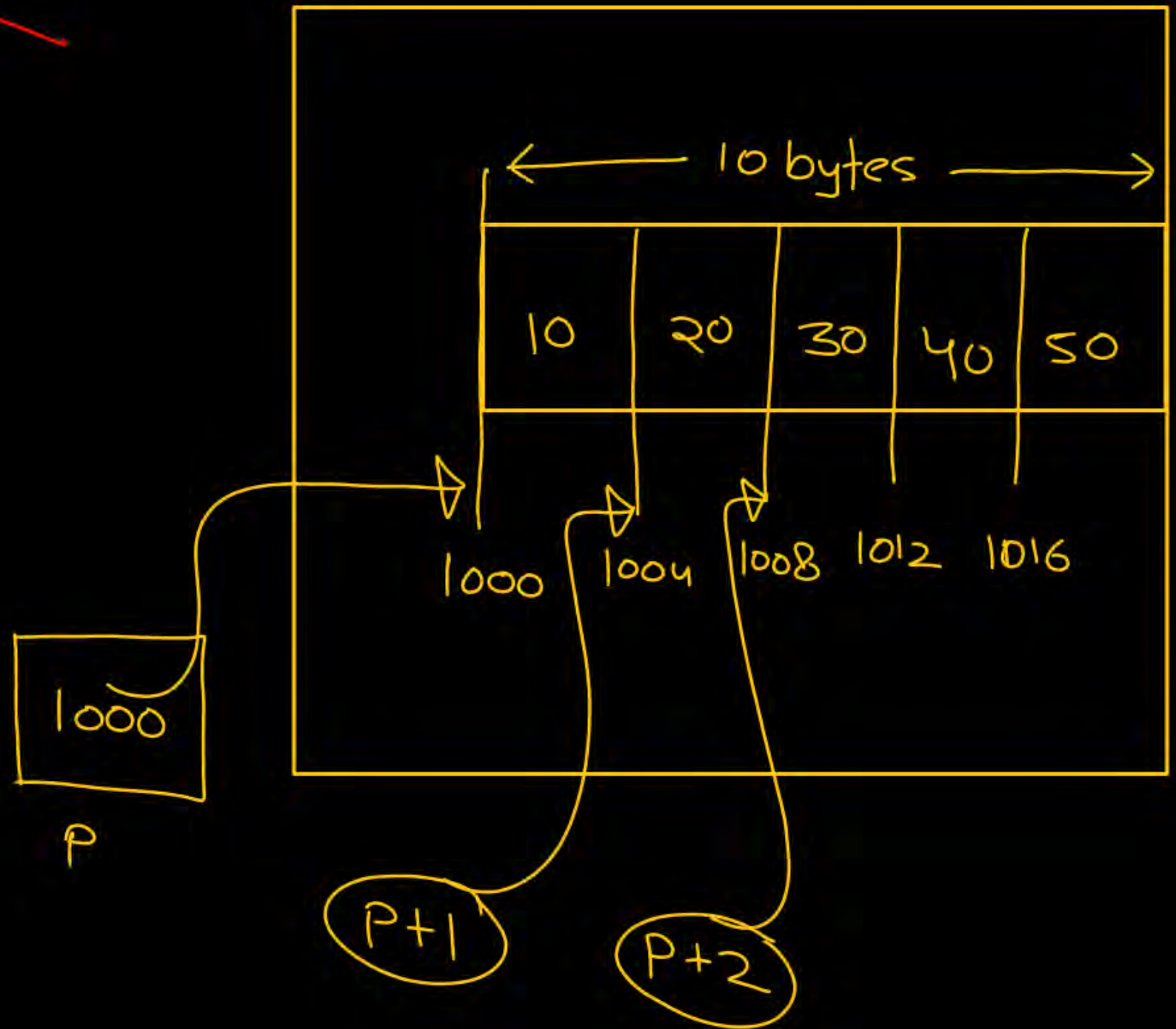
`scanf("/d", p);` 10

`scanf("/d", p+1);` 20

`scanf("/d", p+2);` 30

`scanf("/d", p+3);` 40

`scanf("/d", p+4);` 50





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

repeated

`scanf("/d", P+0);` 10

`scanf("/d", P+1);` 20

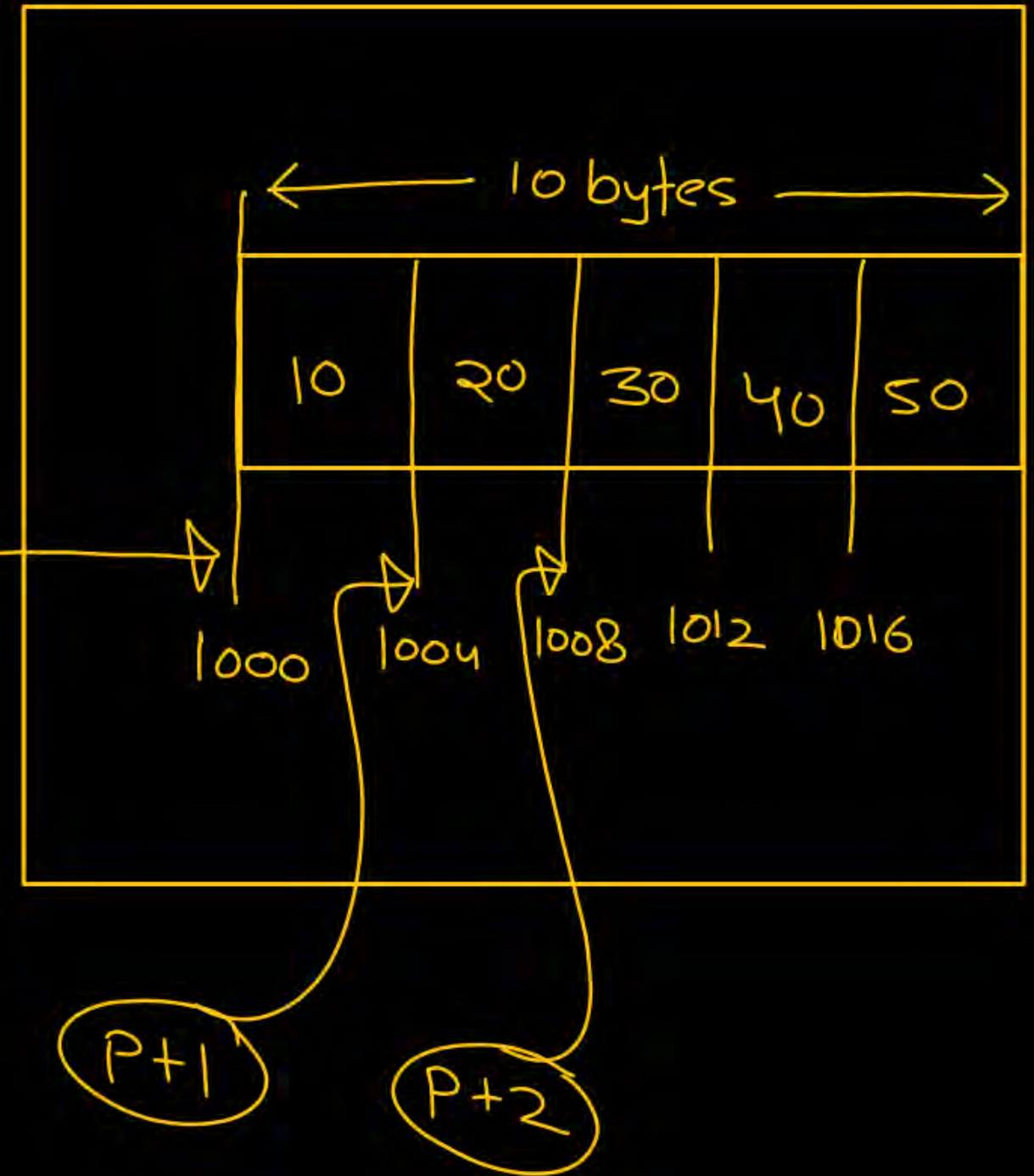
`scanf("/d", P+2);` 30

`scanf("/d", P+3);` 40

`scanf("/d", P+4);` 50

repeat

1000  
P



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

```
for(i=0; i<5; i++)  
scanf("%d", p+i)] 5 values read ✓
```

$p \Rightarrow 1000$

$*p \Rightarrow 10$

`printf("%d", *p);`  $\Rightarrow 10$

$p+1 \Rightarrow 1004$

$*p+1 \Rightarrow 20$

`printf("%d", *(p+1));`  $\Rightarrow 20$

`printf("%d", *(p+2));`  $\Rightarrow 30$

`printf("%d", *(p+3));`  $\Rightarrow 40$

`printf("%d", *(p+4));`  $\Rightarrow 50$

10 ↓

20 ↓

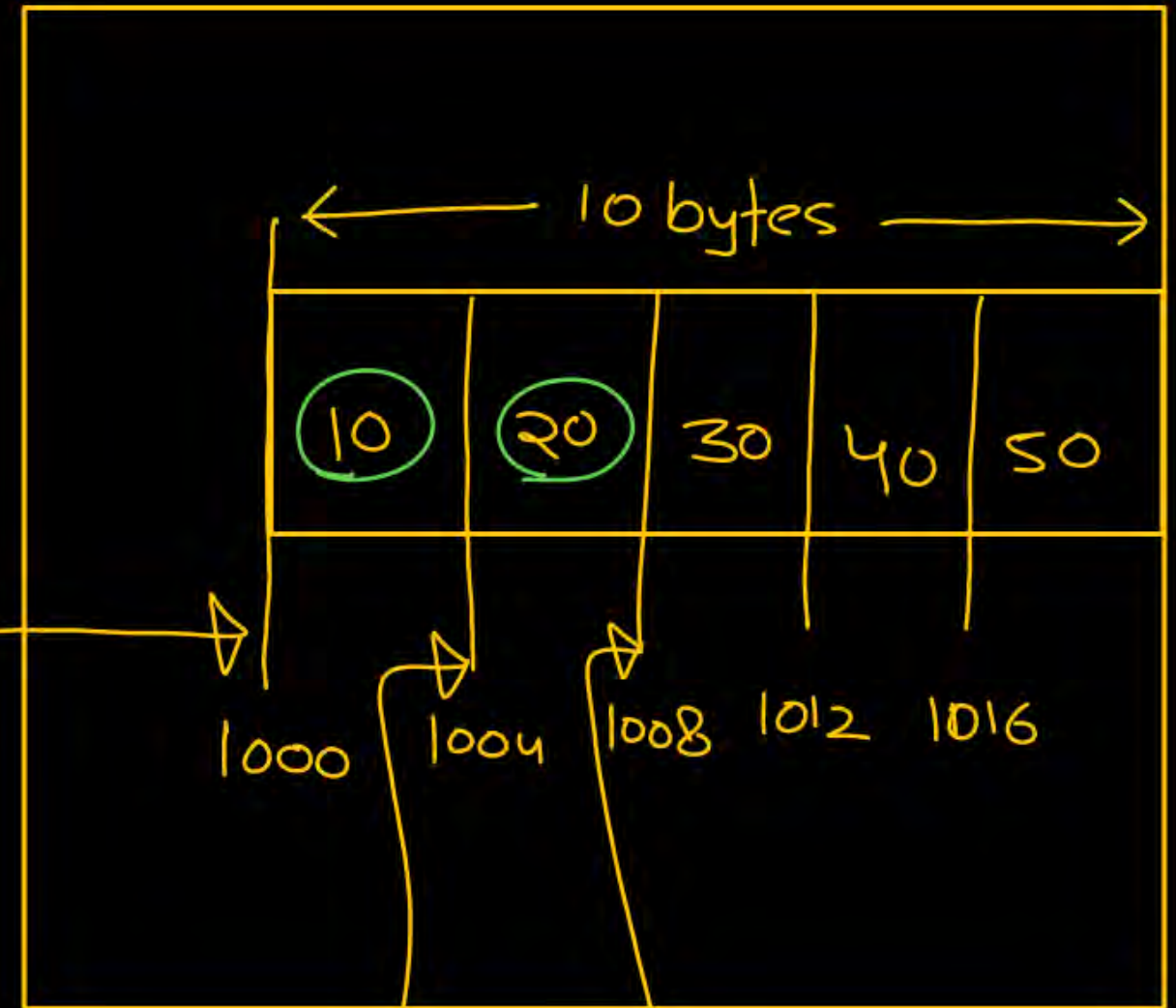
30 ↓

40 ↓

50 ↓

1000

p



P+1

P+2



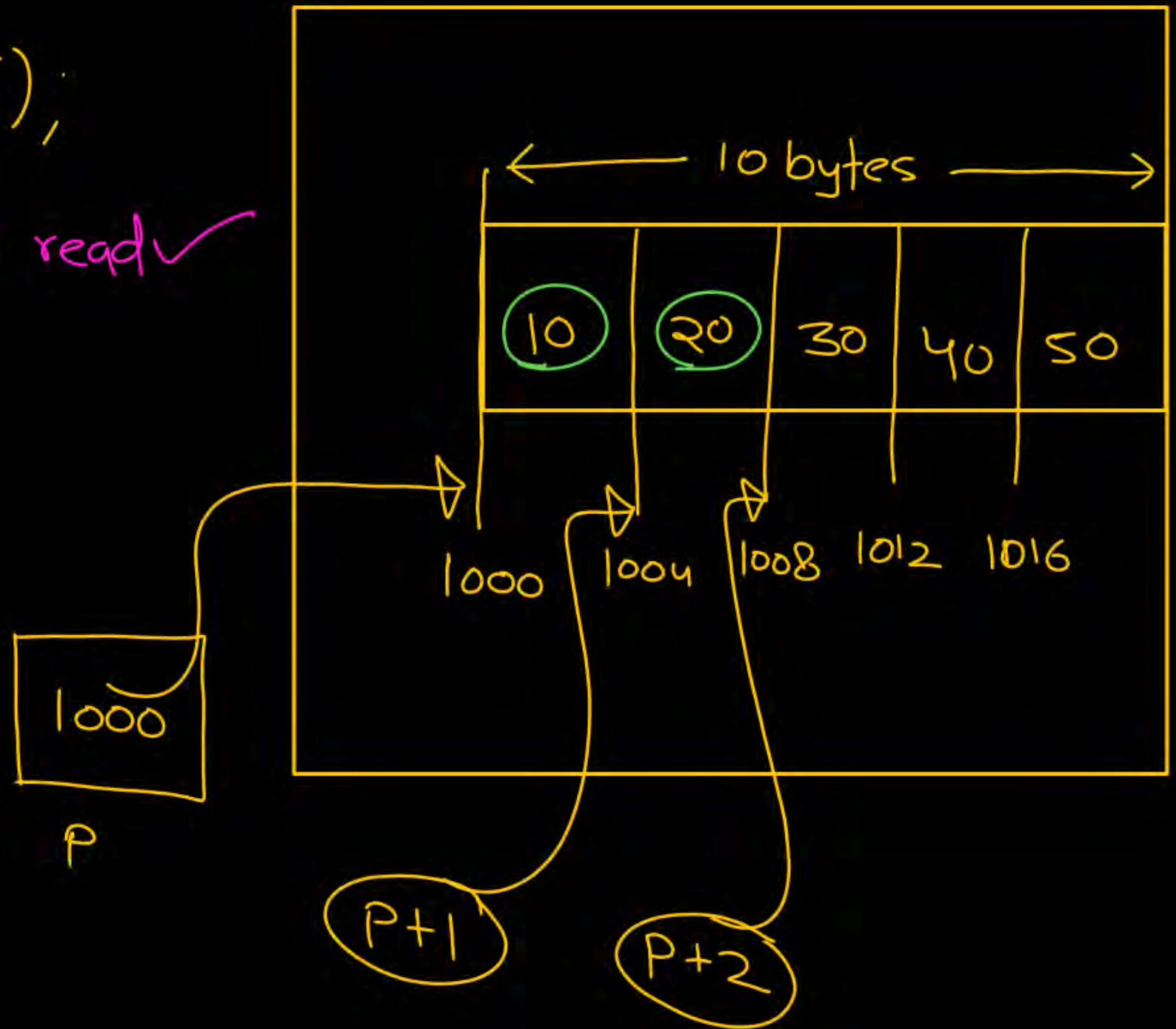
```
int *p = malloc(sizeof(int) * 5);
```

```
for(i=0; i<5; i++)  
scanf("%d", p+i)] 5 values read ✓
```

```
for(i=0; i<5; i++)  
printf("%d", *(p+i));
```

OR

```
for(i=0; i<5; i++)  
printf("%d", p[i]);
```





malloc(1000000);

↙  
available

→ Not available

Starting add. of block

↓  
NULL pointer

```
int N;  
int i, *p;  
printf("Enter the no. of elements");  
scanf("%d", &N);  
  
p = malloc(sizeof(int) * N);  
if (p != NULL) {  
    for (i = 0; i < N; i++)  
        scanf("%d", p + i);  
  
    for (i = 0; i < N; i++)  
        printf("%d", p[i]);  
}
```



## Calloc

2 argument

$\text{calloc}(\text{No. of block}, \text{size of each block})$

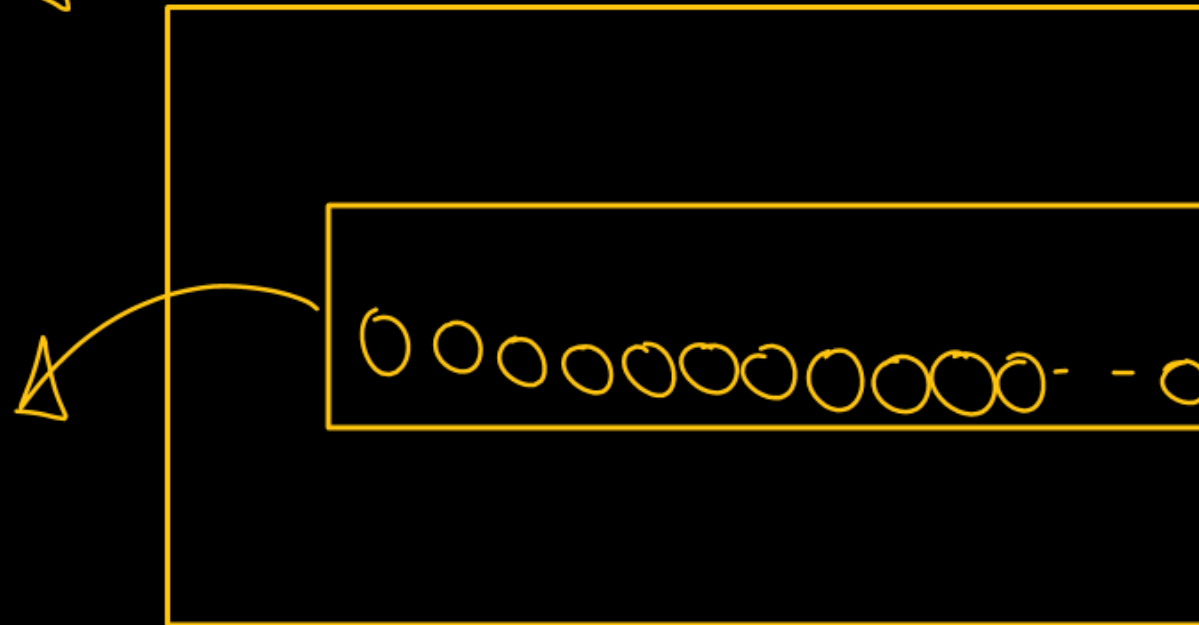
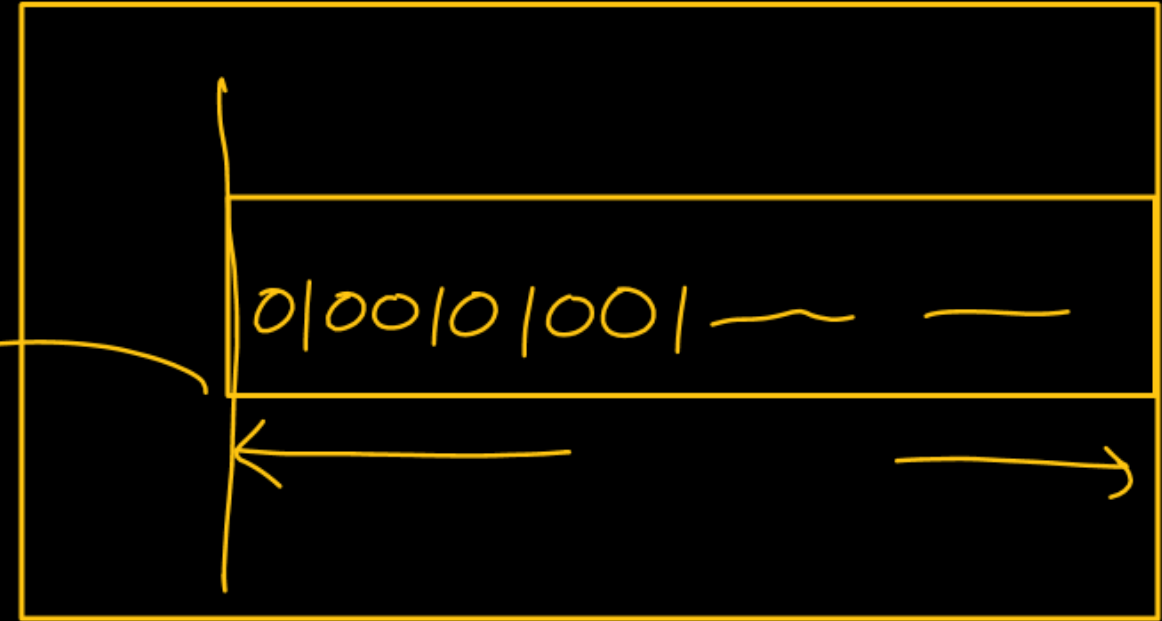
$\text{malloc}(5 \times \text{sizeof(int)}) \Rightarrow \text{calloc}(5, \text{sizeof(int)})$

malloc → ① Search  
② starting address —  
Cheaper

Calloc

① Search  
② 0 → bits  
③ <sup>starting</sup> Add

} more time  
expensive





NULL

realloc  
↙  
again

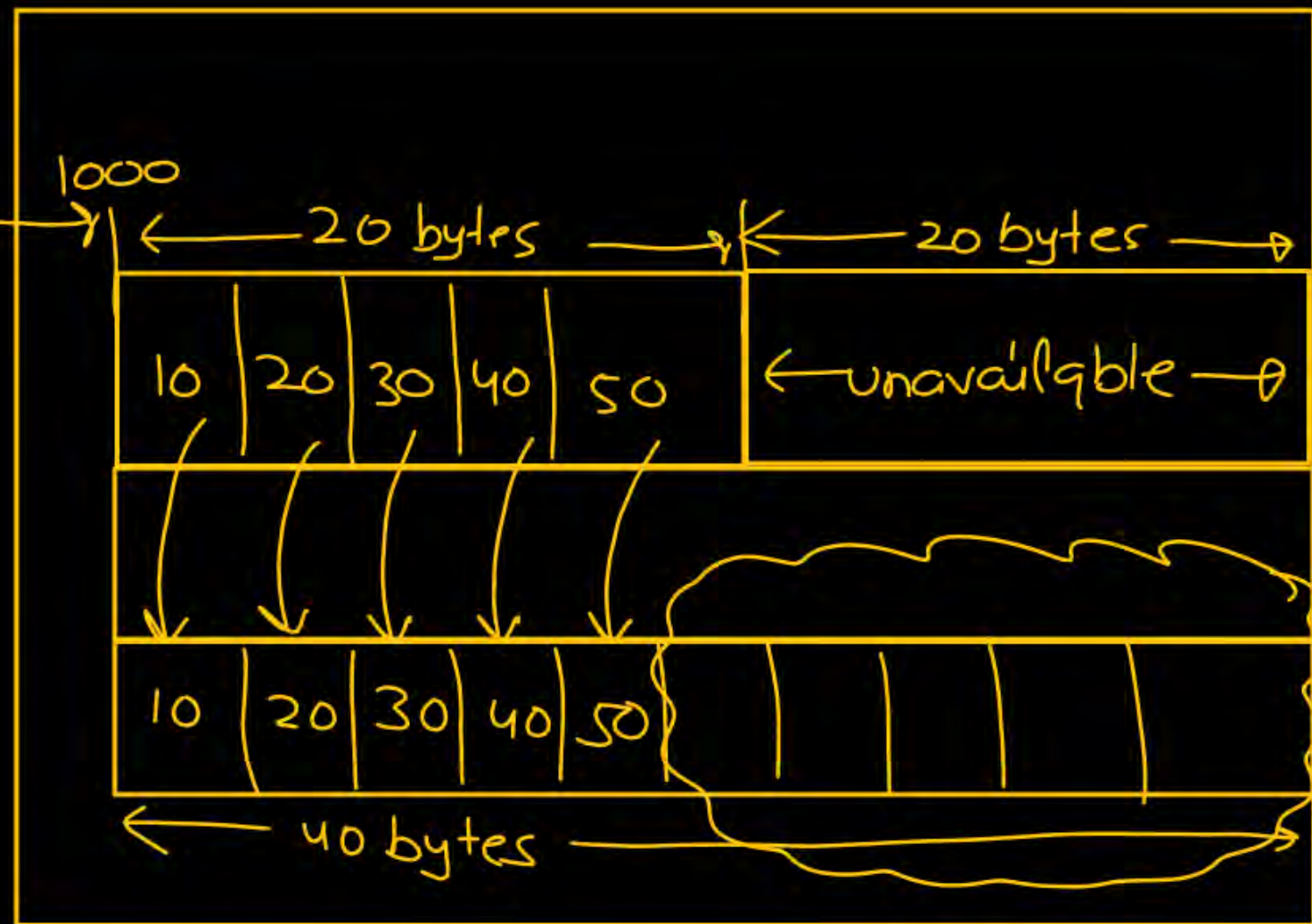
```
int *P = malloc(s * sizeof(int));
```

```
for( _____ )  
    _____
```

```
P = realloc(P, new-size) P
```

```
P = realloc(P, 10 * sizeof(int));
```

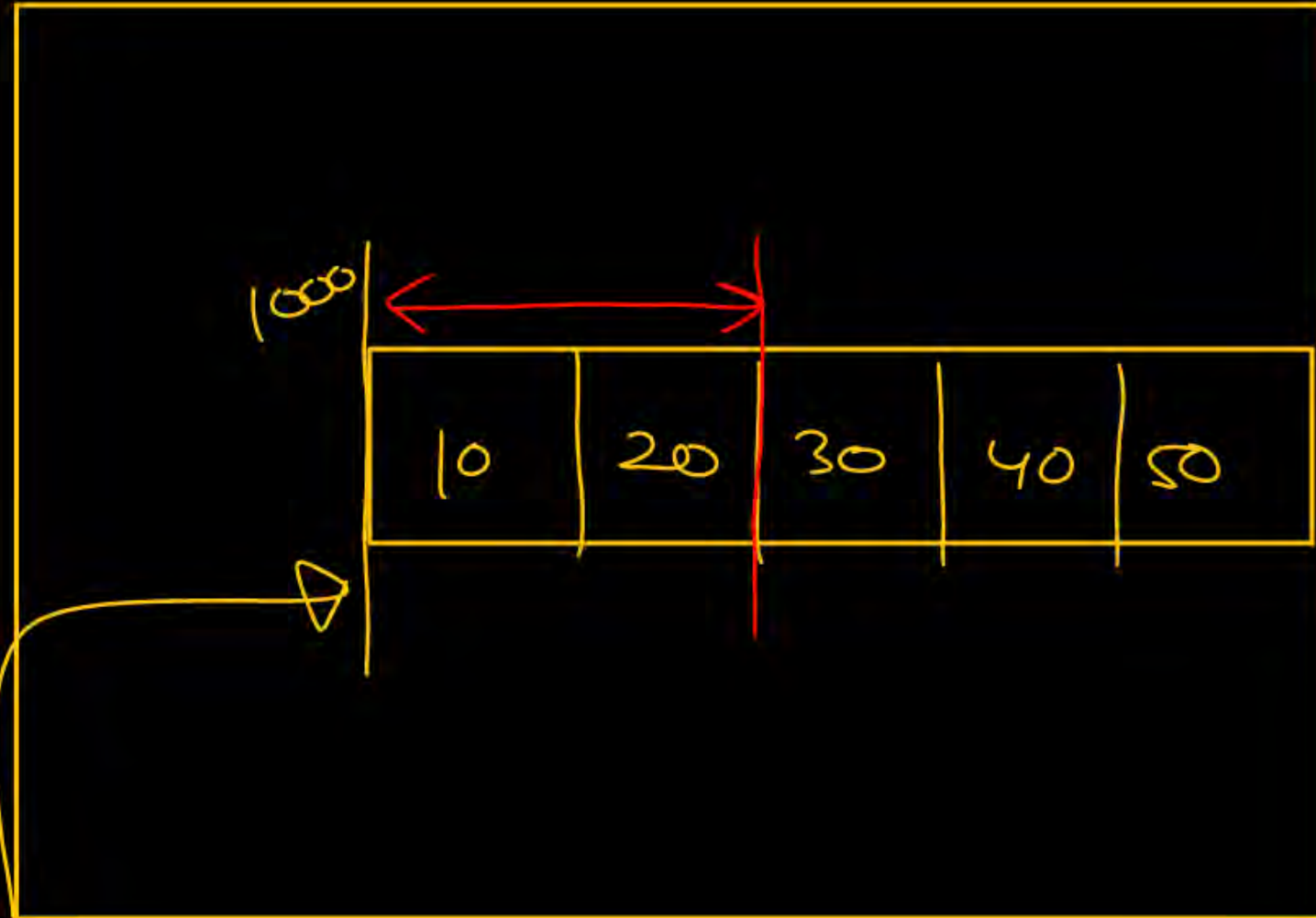
1000



```
int *p = malloc(5 * sizeof(int))
```

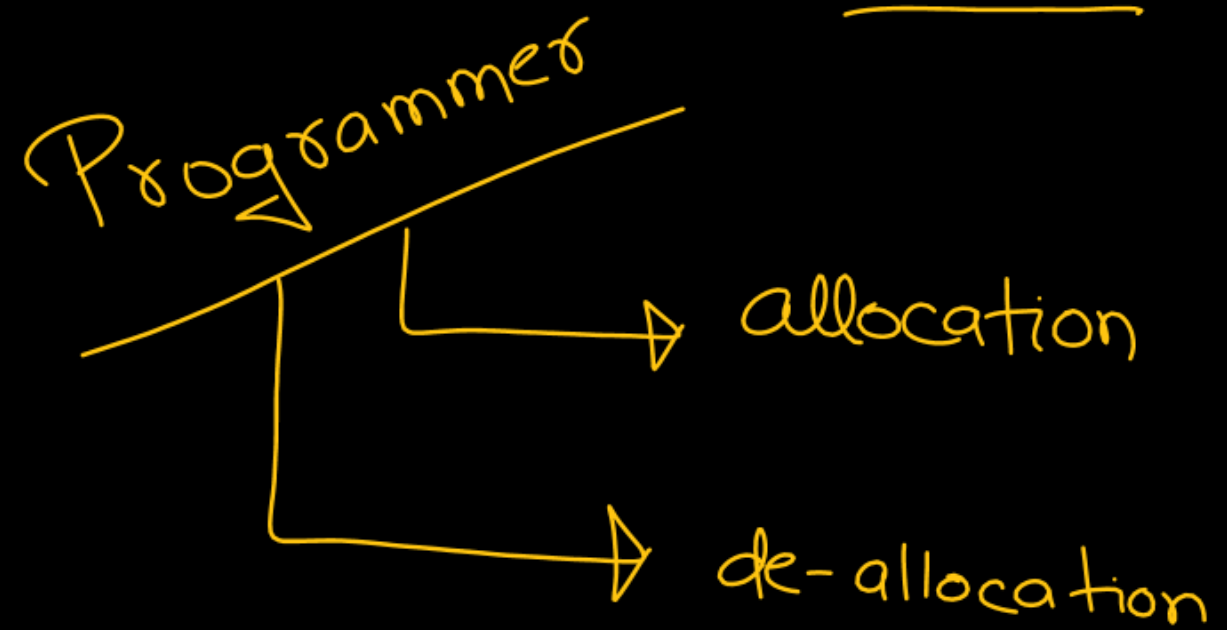
for       

```
p = realloc(p, 2 * sizeof(int));
```





## DMA



Memory  
leakage  
Problem

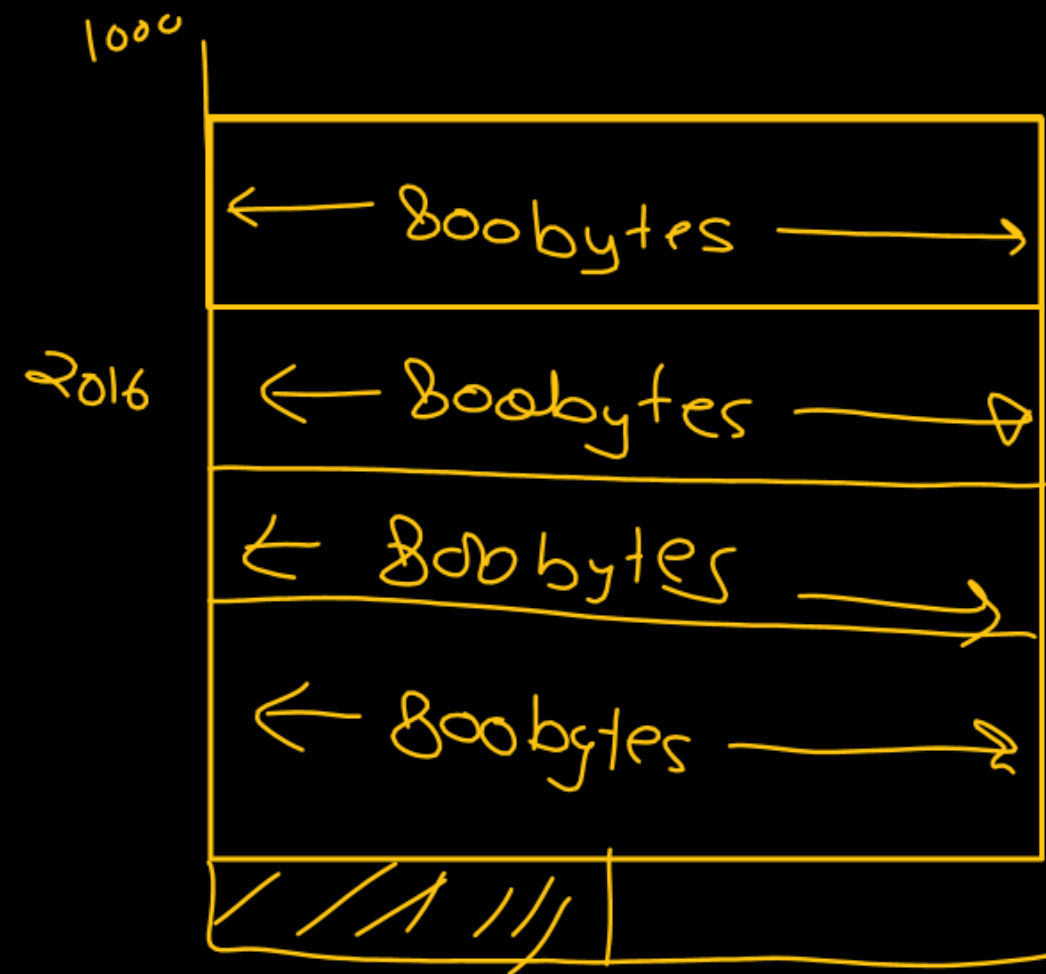
```
void fun() {
```

```
    int *p = malloc(800200 * sizeof(int));
```

```
    //  
    //  
    //  
    return;  
}
```

local  
variable

```
void main() {  
    fun();  
    ✓ fun();  
    fun();  
    fun();  
    fun();  
}
```





free

```
void fun() {
```

```
    int *p = malloc(——);
```

```
    ==
```

```
    free(p);  $\Rightarrow$  de-allocate  
    return;  
}
```

Given an array in which all elements occur even no. of times except one element. find that ele.

int a[] = { 10, 10, 30, 40, 30, 40, 30, 30, 5 }

$$a \wedge a = 0$$

$$0 \wedge a = a$$

$$\underbrace{a \wedge a \wedge a}_{\text{even times}} \Rightarrow a$$

$$\underbrace{10 \wedge 10}_{\text{even times}} \wedge \underbrace{30 \wedge 40 \wedge 30 \wedge 40}_{\text{even times}} \wedge \underbrace{30 \wedge 30}_{\text{even times}} \wedge 5$$

$$0 \wedge 5 = \textcircled{5}$$





C++/Java

String

DSA

