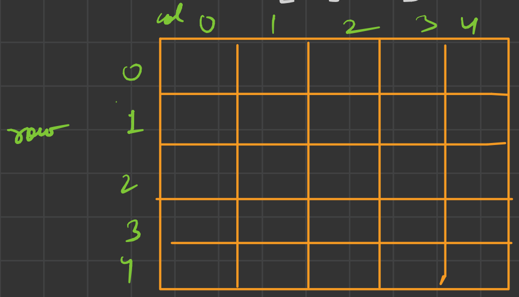
Dynamic Memory Allocation [Part-2]

int arr[5][5]



int arr[10]

or

dynamic

int *arr = new int [10];

↓

Variable / user vale

run time

int *arr = new int[n];

int m → rows

int n → cols

↓

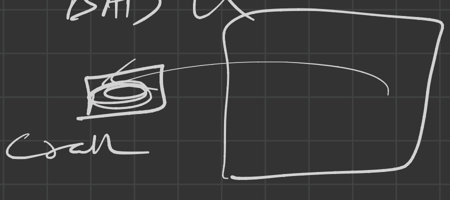
run time → 2D array → ?

int m, n;

cin >> m >> n;

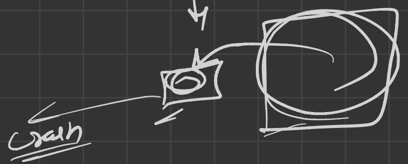
int arr[m][n];

BAD ✗



int arr[n] → ✗ BAD

↓

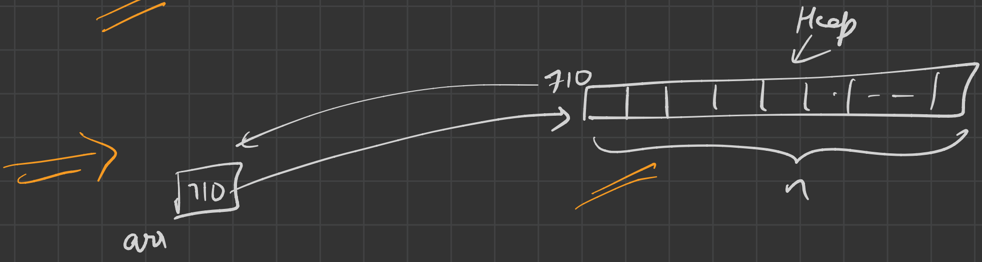


Dynamic Memory

↳ 2D array

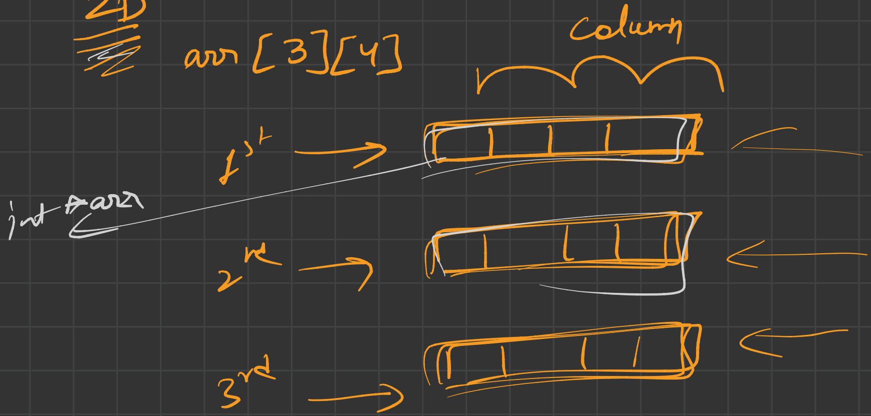
User i/p
[runtime]

```
int * arr = new int [n];
```



2D

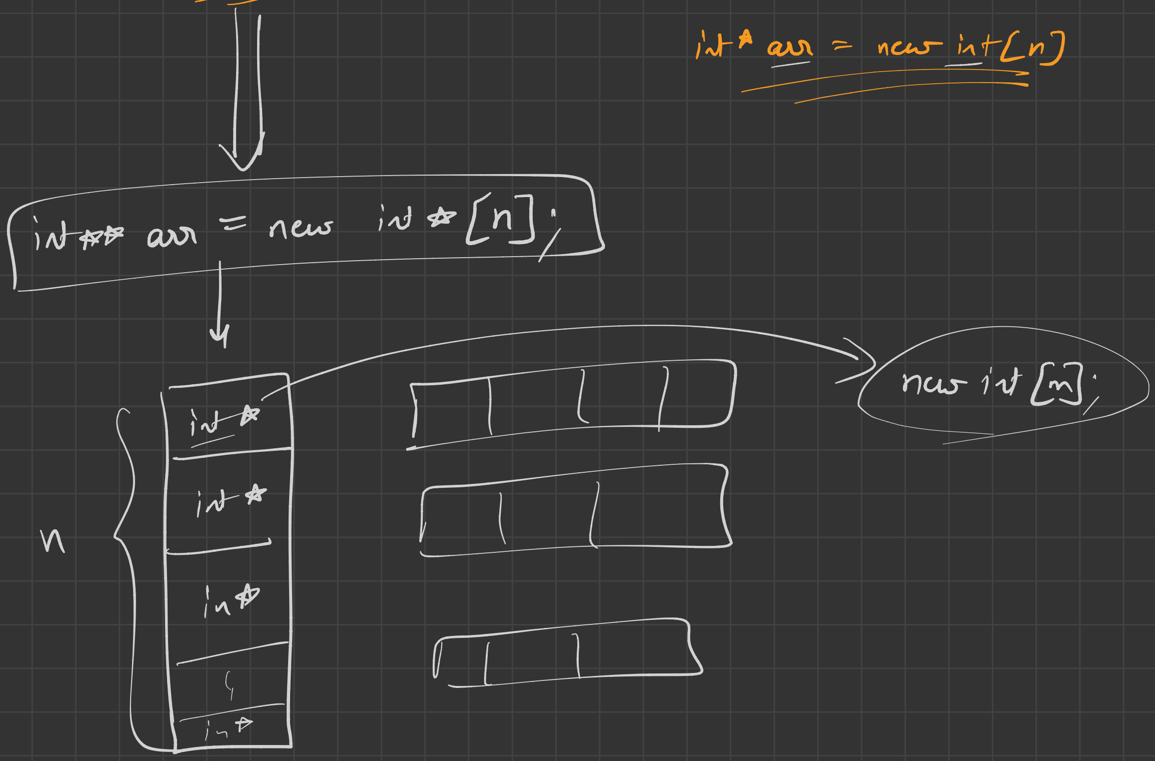
arr [3][4]



2D array

1D arr

int* arr = new int[n]

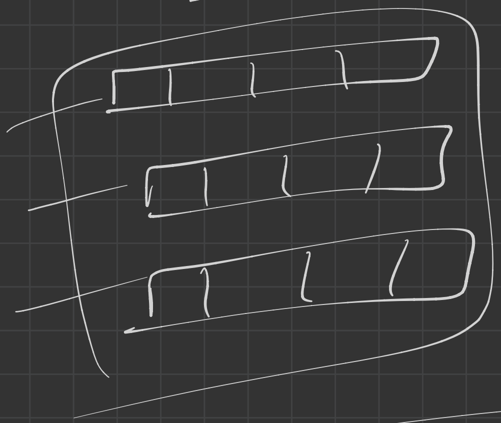


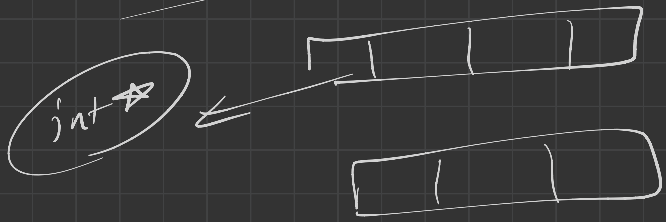
2D

1D

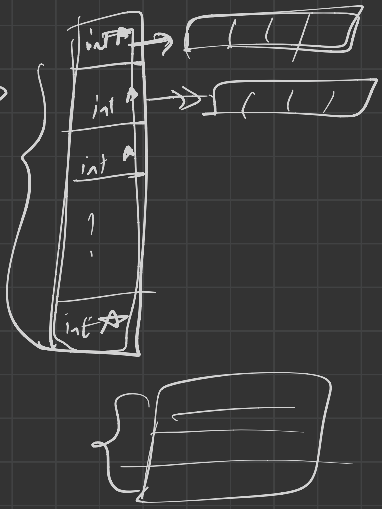
```
int* arr = new int[n]
```

```
arr[n][n];
```





```
int** arr = new int*[n];
```



```
For (            )
{
    arr[i] = new int[n];
}
```

2D creation done

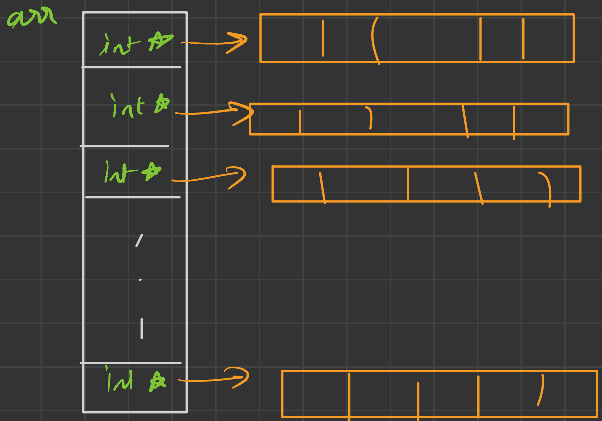
→ Heap → memory leak

↳ Kaam jab hojaye

↓

memory free (manually)

2D :- Visualisation



Home work :- Jaggered Array

↳ dynamic memory Allocation