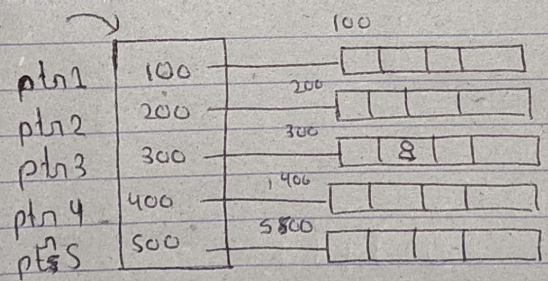
⇒ Static Memory Allocation: when we allocate memory at compile time.

Day - 71
Dynamic Memory Allocation (2D - Array)

⇒ For 1-D array → int *ptr = new int[4];
⇒ 2D-array is the combination of 1-D array.
⇒ So, we can create 1-D array —

```
int *ptr2 = new int[4]
int *ptr3 = new int[4]
int *ptr4 = new int[4]
int *ptr5 = new int[5]
```

⇒ But what if we want 100 rows in our 2-D array then we can't do above method —
⇒ So, for this we can create a array that stores addresses.



⇒ ptr is a double pointer.

⇒ So, by using ptr, we can access any row.
⇒ Let, we want to access 8 —
then, *(*(ptr+2)+1) ⟶ ptr[2][1]
⇒ Both are same.

Date : / /
Page:

⇒ For creating 2D array :

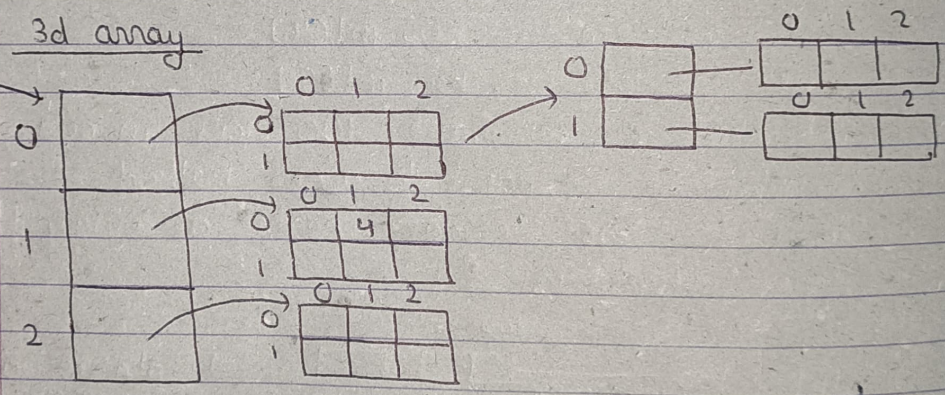
```
int ** ptr = new int *[n];
for (int i = 0; i < n; i++)
    ptr[i] = new int [m];
for( int i=0; i<n; i++)
    for(int j=0; j<m; j++)
        cin >> ptr[i][j];
```

⇒ For releasing memory :

```
for( int i= 0; i<n; i++)
    delete [] ptr[i];
```

⇒ `delete [] ptr;`

3d array



⇒ For accessing 4,

ptr[1][0][1]

```
*(*(*(ptr + 1) + 0) + 1)
```

⇒ To create

```
int *** ptr = new int **[1]
```

```
for( i=0; i<L; i++){
    ptr[i] = new int *[B];
    for( j=0; j< B; j++)
        ptr[i][j] = new int[H];
}
```