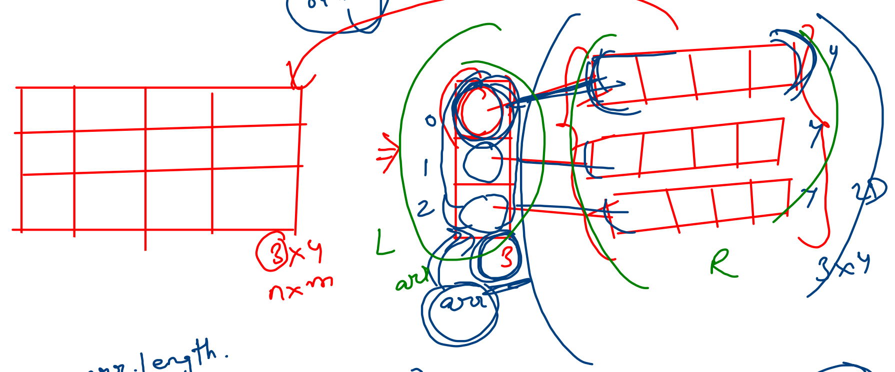
2D. → 1D.

Memory. 2D → concept → actually 1D.

@man → character (string)

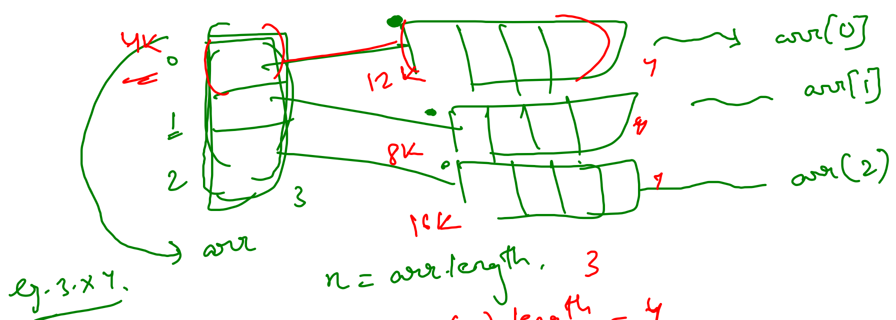
2D?



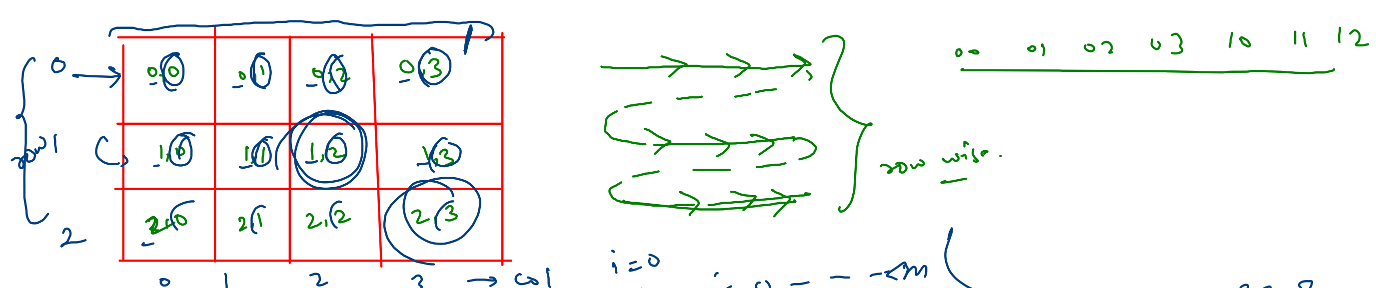
1D → n = arr.length.

2D → int n = arr.length → 3
m = arr[0].length → 4

2D arr



n = arr.length. 3
m = arr(0).length = 4



row → i
col → j

r = 0 = i ↳ 0, 1, 2, 3
r = 1 = i ↳ j = 0, 1, 2, 3
r = 2 = i j = 0, 1, 2, 3.

i = 0 ↳ j = 0 ---- < m
i = 1 j = 0 ---- < m
i = 2 j = 0 ---- < m

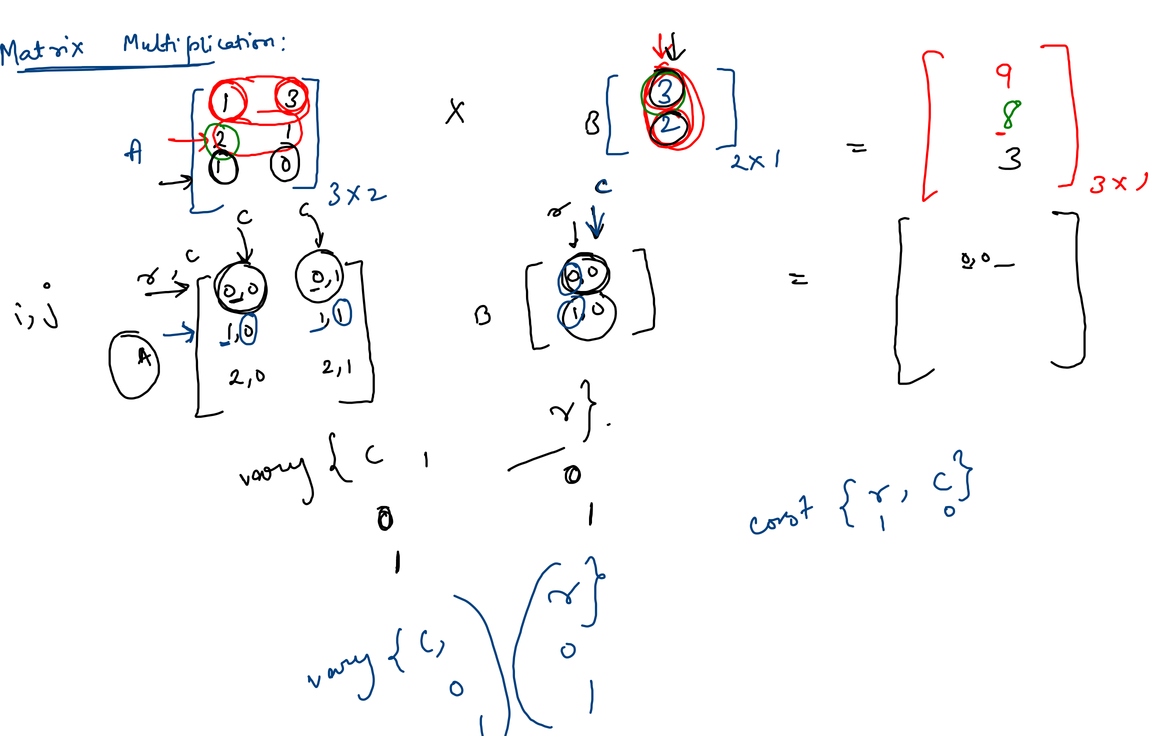
for(i = 0 → < n)
{ for(j = 0 → < m)
{
} }

(i, j)

arr[1][2]

2, 3

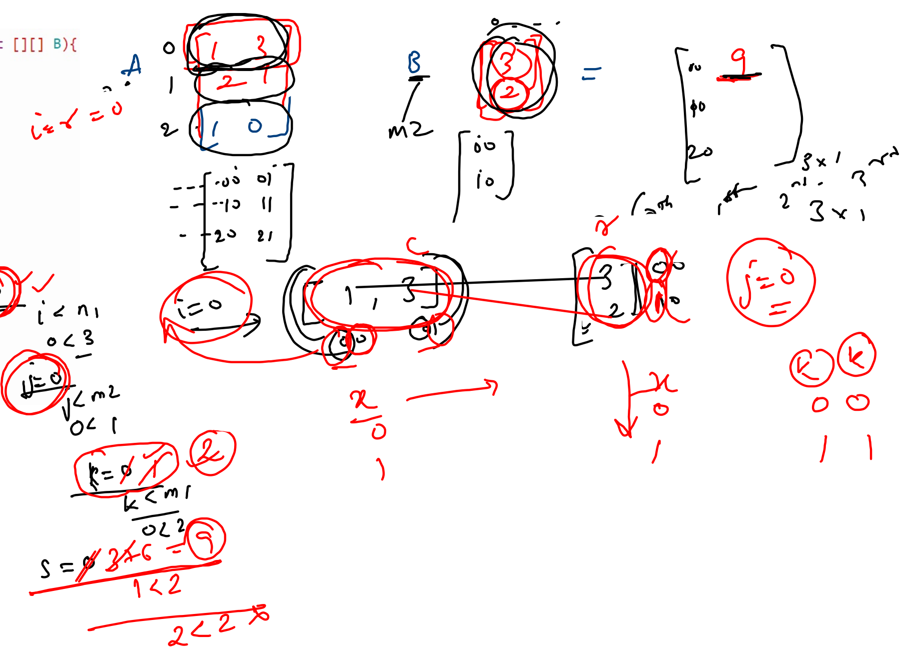
Matrix Multiplication:




```java
public static int [][] matrixMul(int [][]A, int [][] B){
    int n1 = A.length;
    int m1 = A[0].length;
    int n2 = B.length;
    int m2 = B[0].length;

    int [][] C = new int[n1][m2];

    for(int i = 0; i<n1; i++){
        for(int j = 0; j<m2; j++){
            int sum = 0;
            for(int k = 0; k<m1; k++){
                sum += A[i][k] * B[k][j];
            }
            C[i][j] = sum;
        }
    }
    return C;
}
```



A(0)(0) = 1 × B(0)(0) = 3 → 3
A(0)(1) = 3 × B[1][0] = 2 → 6