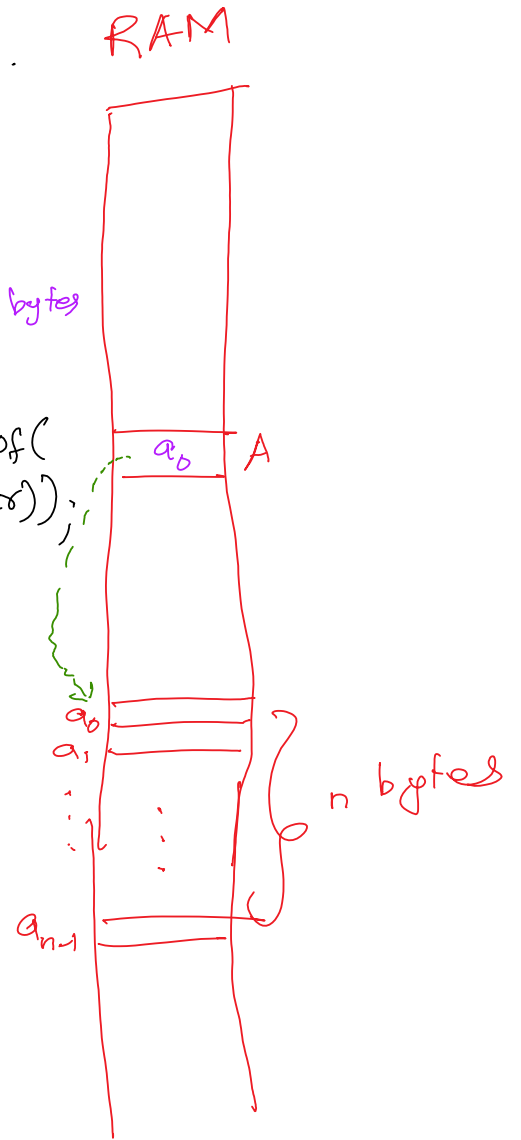


C pointers

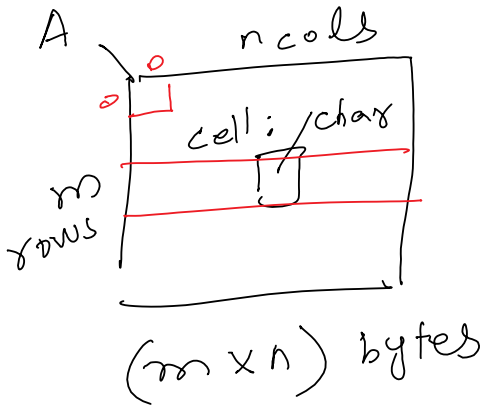
Tuesday, October 9, 2018 9:36 PM

C pointers :

```
void f( ) {  
    char *A; ←  
    :  
    :  
    A = (char *) malloc( n * sizeof(  
    :                               char) );  
    :  
    :  
}
```



Concept 2) Two Dimensional Array:



→ char A[m][n];

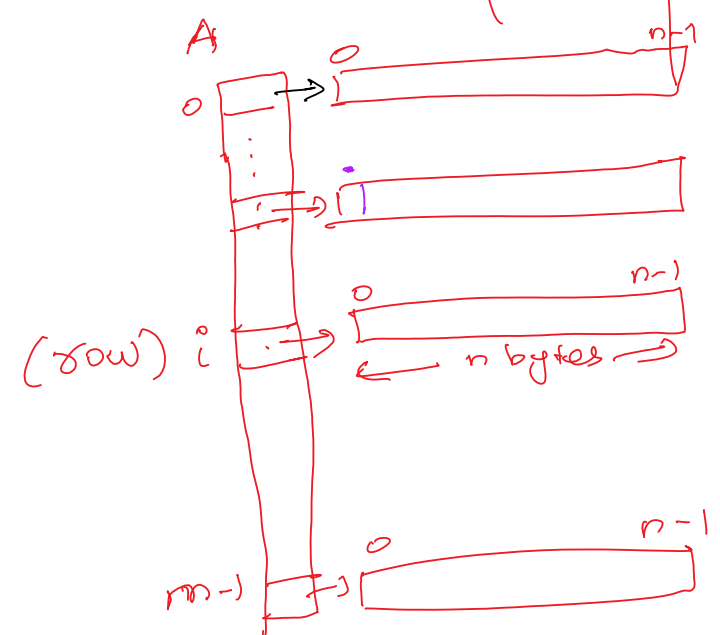
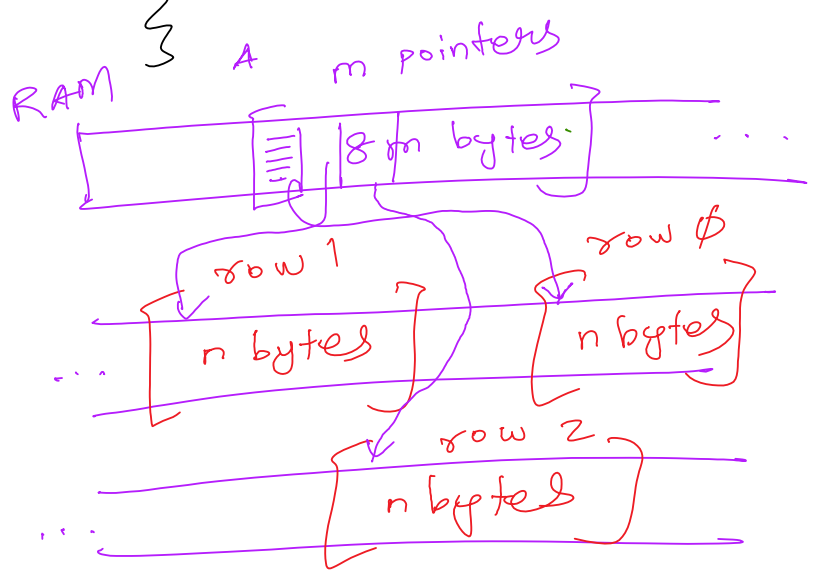
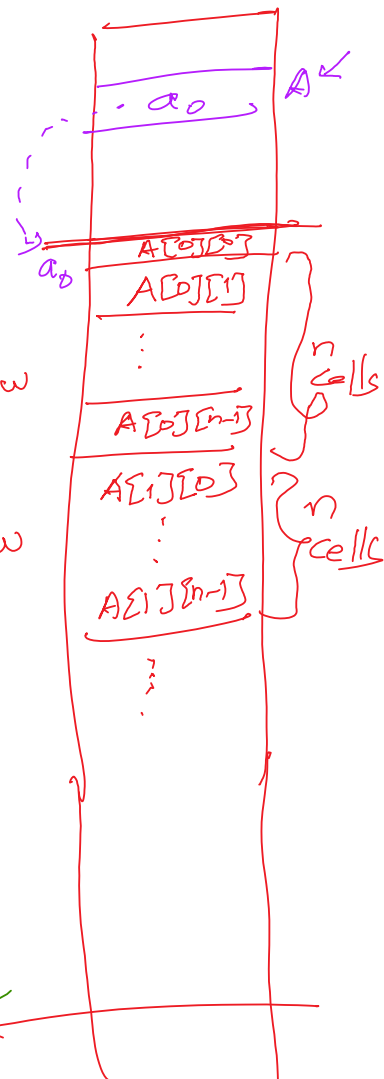
A

w/ pointers:

```
char **A;
A = (char **) malloc(m * sizeof(char *));
for (i = 0; i < m; i++) {
    A[i] = (char *) malloc(n * sizeof(char));
}
```

m x n bytes

m x pointer

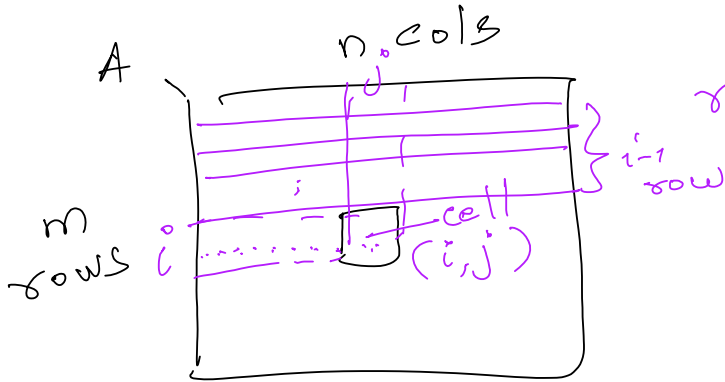


2-D array implementations:

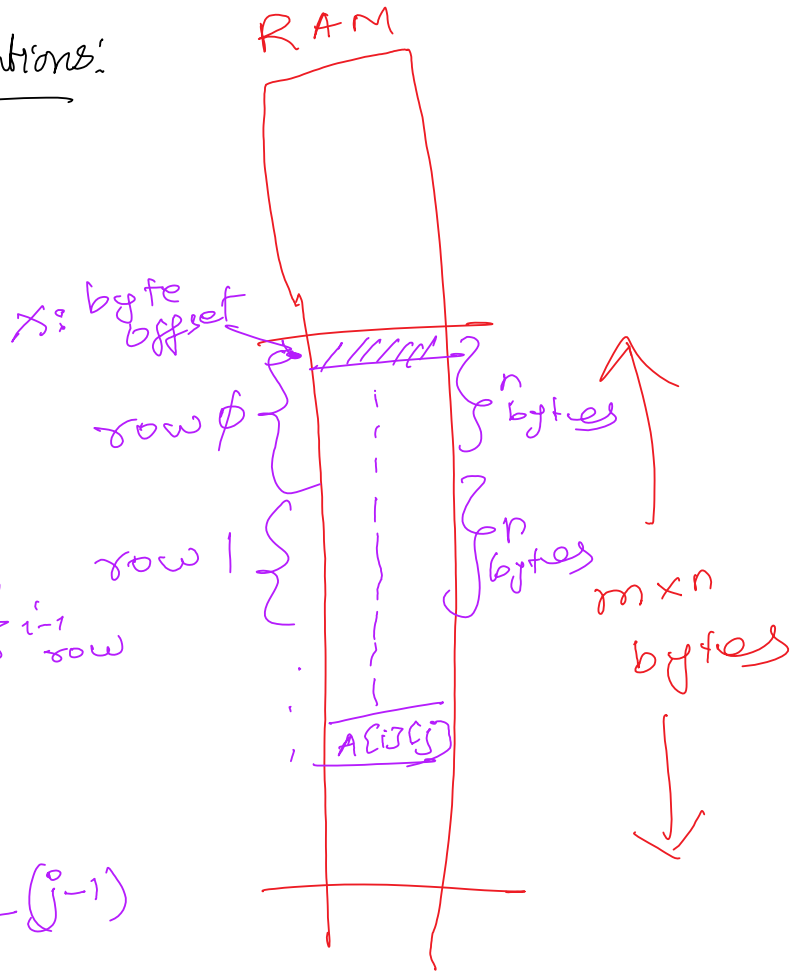
1) char A[m][n]; ←

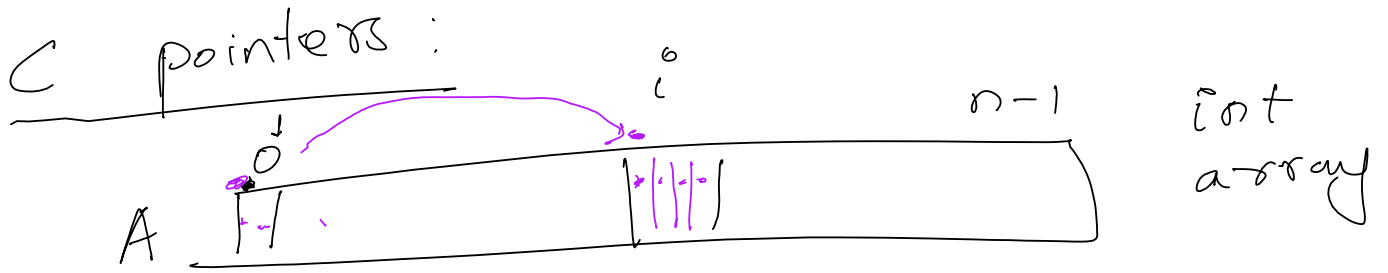
2) char **A; ←

3) char *A; ←



$m \times n$ bytes $x + (i-1)n + (j-1)$





```

int
char * A;
A = (int *) malloc ( sizeof (int) * n );
// init .
:
:
A == & ( A [ 0 ] )
    
```

address of
data

assert (A == & (A [0]));

// how to access the i^{th} element?

$$A[i] = * (A + i)$$

address of A[0]

MPI: $\text{int } x; \dots 1 \text{ integer}$

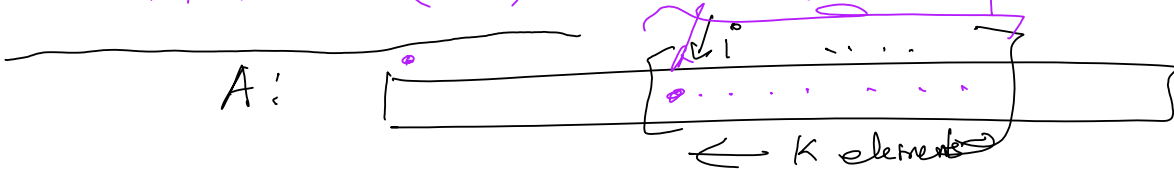
$\text{MPI_send}(\text{send-buf}, \text{send-count}, \text{send-type} \dots)$

↓
address

$\text{MPI_send}(\&(x), 1, \text{MPI_INT} \dots)$

$\text{int } *x;$
// $x =$ allocate space for n ints.

$\text{MPI_send}(x, n, \text{MPI_INT} \dots)$



$\text{MPI_send}(\&(A[i]), K, \text{MPI_INT}, \dots)$