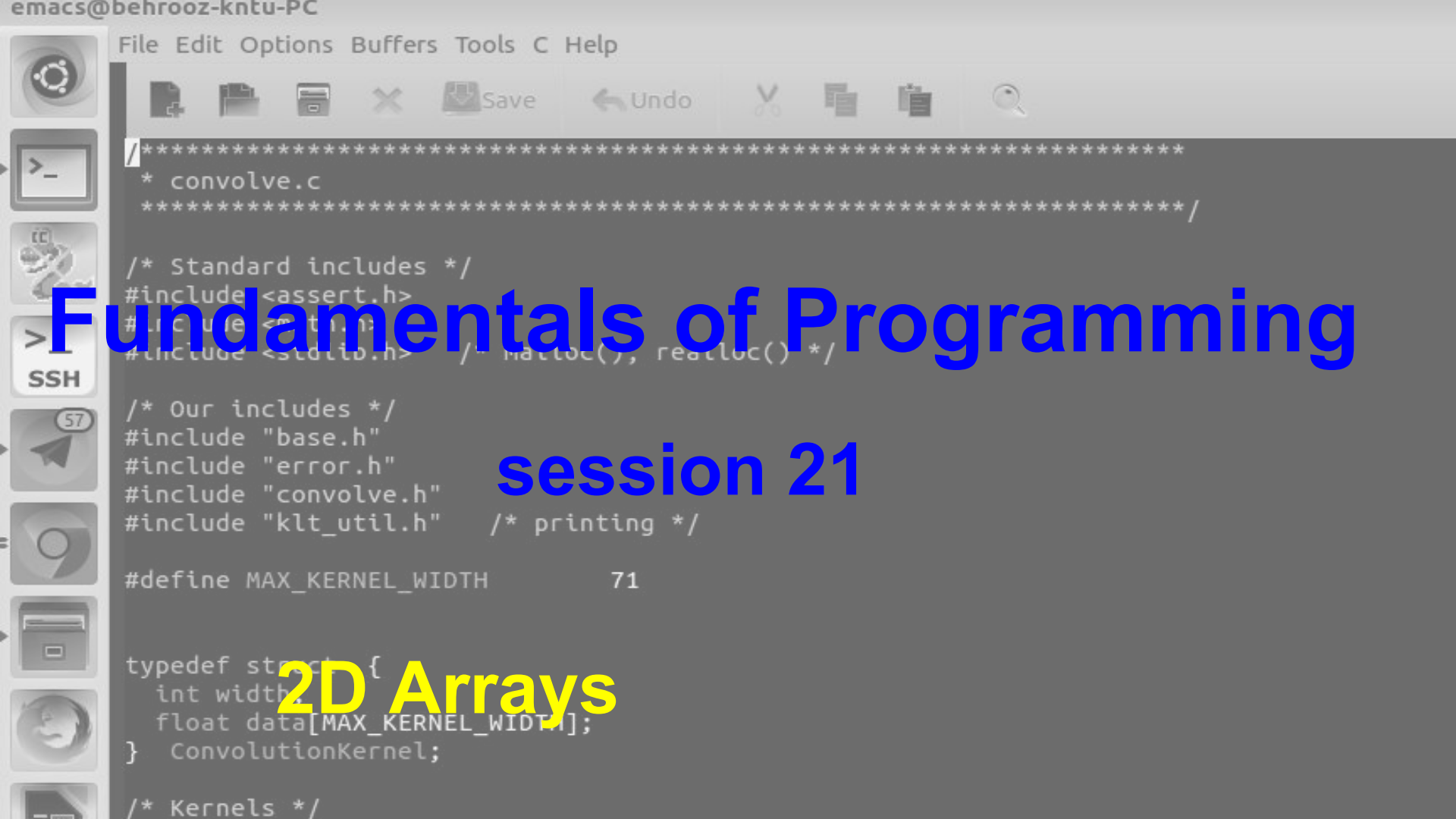

emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
/* convolve.c */
/* Standard includes */
#include <assert.h>
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
Fundamentals of Programming
session 21
2D Arrays

2D arrays

0	0	0	0	0	0
0	1	2	3	4	5
0	2	4	6	8	10
0	3	6	9	12	15
0	4	8	12	16	20

- tabular data
- rows and columns

2D arrays

0	0	0	0	0	0
0	1	2	3	4	5
0	2	4	6	8	10
0	3	6	9	12	15
0	4	8	12	16	20
0	5	10	15	20	25

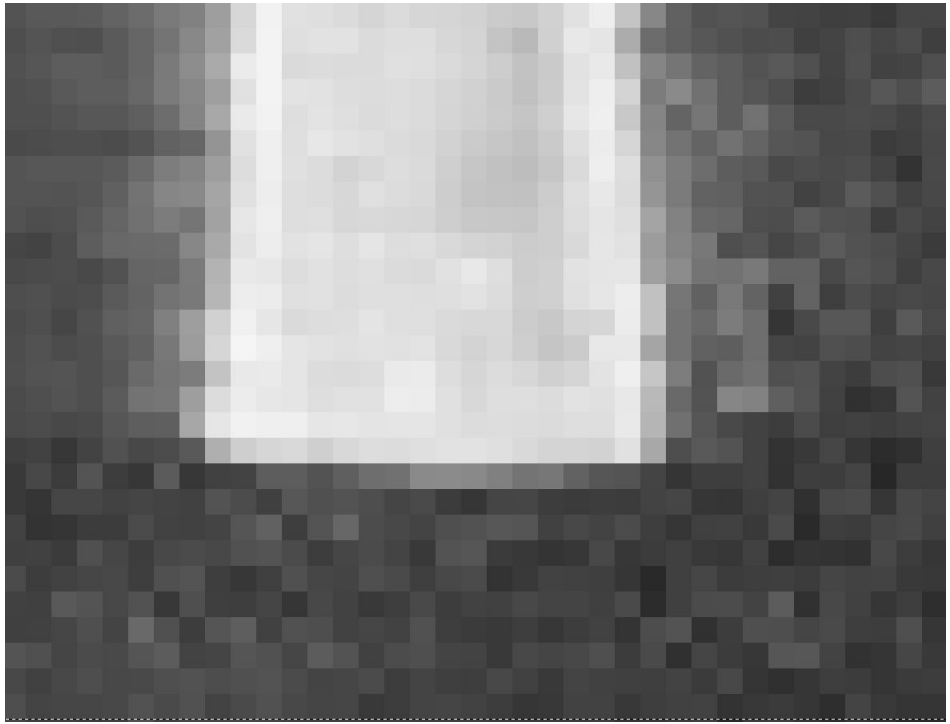
$$A = \begin{pmatrix} 3 & -5 & 4 \\ 9 & 8 & -7 \\ -6 & 4 & 2 \end{pmatrix}, B = \begin{pmatrix} -2 & -1 & 1 \\ 5 & -7 & 6 \\ 9 & 3 & 2 \end{pmatrix}$$

https://advancedmathclubsk.weebly.com/matrices.html

2D arrays

https://hiveminer.com/Tags/desert,isfahan/Recent

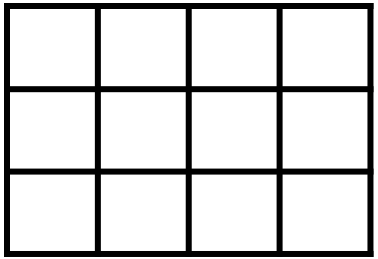
https://hiveminer.com/Tags/desert,isfahan/Recent



https://hiveminer.com/Tags/desert,isfahan/Recent

Defining 2D arrays

```
int a[3][4];
```



Initializing 2D arrays

```
int a[3][4] = {{1,3,5,7}, {2,4,6,8}, {4,11,-1,7}};
```

1	3	5	7
2	4	6	8
4	11	-1	7

Initializing 2D arrays

```
int a[3][4] = {{1, 3, 5, 7},
               {2, 4, 6, 8},
               {4,11,-1,7}};
```

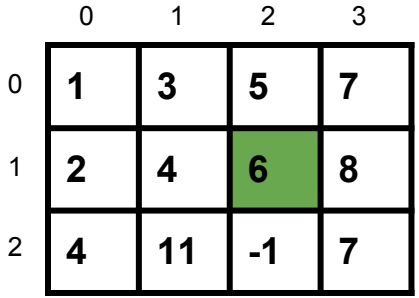
```
printf("%d\n", a[1][2]);
```

	0	1	2	3
0	1	3	5	7
1	2	4	6	8
2	4	11	-1	7

2D arrays

```
int a[3][4] = {{1, 3, 5, 7},
               {2, 4, 6, 8},
               {4,11,-1,7}};
```

```
printf("%d\n", a[1][2]);
```



2D arrays

```
int a[3][4] = {{1, 3, 5, 7},
               {2, 4, 6, 8},
               {4,11,-1,7}};
```

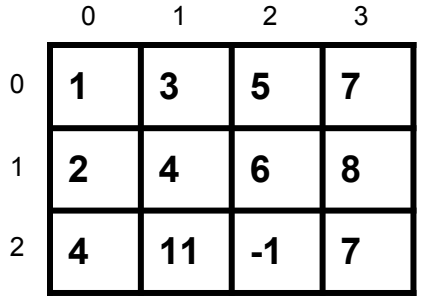
```
printf("%d\n", a[i][j]);
```

	0	1	2	3
0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

2D arrays

```
int a[3][4] = {{1, 3, 5, 7},
               {2, 4, 6, 8},
               {4,11,-1,7}};
```

```
printf("%d\n", a[0][0]);
```



	0	1	2	3
0	1	3	5	7
1	2	4	6	8
2	4	11	-1	7

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    printf("%d\n", a[i][2]);

  return 0;
}
```

2darray.c

	0	1	2	3
0	**1**	**3**	**5**	**7**
1	**2**	**4**	**6**	**8**
2	**4**	**11**	**-1**	**7**

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    printf("%d\n", a[i][2]);

  return 0;
}
```

2darray.c

	0	1	2	3
0	**1**	**3**	**5**	**7**
1	**2**	**4**	**6**	**8**
2	**4**	**11**	**-1**	**7**

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
      printf("%d\n", a[i][j]);

  return 0;
}
```

2darray2.c

	0	1	2	3
0	**1**	**3**	**5**	**7**
1	**2**	**4**	**6**	**8**
2	**4**	**11**	**-1**	**7**

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
      printf("%d\n", a[i][j]);

  return 0;
}
```

2darray2.c

```
$ gcc 2darray2.c  && ./a.out
1
3
5
7
2
4
6
8
4
11
-1
7
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int j = 0; j < 4; j++)
    for (int i = 0; i < 3; i++)
      printf("%d\n", a[i][j]);

  return 0;
}
```

2darray3.c

	0	1	2	3
0	**1**	**3**	**5**	**7**
1	**2**	**4**	**6**	**8**
2	**4**	**11**	**-1**	**7**

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int j = 0; j < 4; j++)
    for (int i = 0; i < 3; i++)
      printf("%d\n", a[i][j]);

  return 0;
}
```

2darray3.c

```
$ gcc 2darray3.c  && ./a.out
1
2
4
3
4
11
5
6
-1
7
8
7
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

  putchar('\n');


  return 0;
}
```

2darray4.c

	0	1	2	3
0	1	3	5	7
1	2	4	6	8
2	4	11	-1	7

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

  putchar('\n');


  return 0;
}
```

2darray4.c

```
$ gcc 2darray4.c  && ./a.out
  1,  3,  5,  7,  2,  4,  6,  8,  4, 11, -1,  7,
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray5.c

	0	1	2	3
0	1	3	5	7
1	2	4	6	8
2	4	11	-1	7

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

```
$ gcc 2darray5.c  && ./a.out
  1,  3,  5,  7,
  2,  4,  6,  8,
  4, 11, -1,  7,
```

2darray5.c

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3},
                 {2, 4, 6, 8},
                 {4}};


  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray6.c

```
$ gcc 2darray6.c  && ./a.out
  1,  3,  0,  0,
  2,  4,  6,  8,
  4,  0,  0,  0,
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {

  int a[3][4] = {1,3,5,7,2,4,6,8,4,11,-1,7};

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray7.c

```
$ gcc 2darray7.c  && ./a.out
  1,  3,  5,  7,
  2,  4,  6,  8,
  4, 11, -1,  7,
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {

  int a[3][4] = {1,3,5,7,2,4};

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray8.c

```
$ gcc 2darray8.c  && ./a.out
  1,  3,  5,  7,
  2,  4,  0,  0,
  0,  0,  0,  0,
```

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  for (int i = 0; i < 3; i++)
    a[i][2] = 0;

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray9.c

	0	1	2	3
0	**1**	**3**	**5**	**7**
1	**2**	**4**	**6**	**8**
2	**4**	**11**	**-1**	**7**

Working with 2D arrays

```c
#include <stdio.h>

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  for (int i = 0; i < 3; i++)
    a[i][2] = 0;

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 4; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

2darray9.c

```
$ gcc 2darray9.c  && ./a.out
  1,  3,  0,  7,
  2,  4,  0,  8,
  4, 11,  0,  7,
```

Store times table in a 2D array

```c
#include <stdio.h>

#define M 10
#define N 10

int main() {
  int a[M][N];

  // write your code here

  for (int i = 0; i < M; i++) {
    for (int j = 0; j < N; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

timestable1.c

	0	1	2	3
0	1	2	3	4
1	2	4	6	8
2	3	6	9	12

Store times table in a 2D array

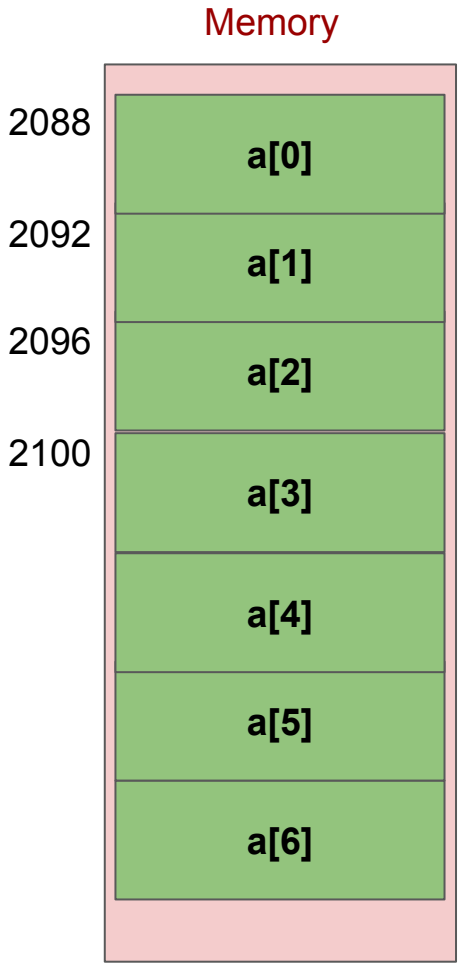
```c
#include <stdio.h>

#define M 10
#define N 10

int main() {
  int a[M][N];

  for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
      a[i][j] = (i+1)*(j+1);

  for (int i = 0; i < M; i++) {
    for (int j = 0; j < N; j++)
      printf("%3d,", a[i][j]);

    putchar('\n');
  }

  return 0;
}
```

timestable1.c

```
$ gcc timestable1.c && ./a.out
  1,  2,  3,  4,  5,  6,  7,  8,  9, 10,
  2,  4,  6,  8, 10, 12, 14, 16, 18, 20,
  3,  6,  9, 12, 15, 18, 21, 24, 27, 30,
  4,  8, 12, 16, 20, 24, 28, 32, 36, 40,
  5, 10, 15, 20, 25, 30, 35, 40, 45, 50,
  6, 12, 18, 24, 30, 36, 42, 48, 54, 60,
  7, 14, 21, 28, 35, 42, 49, 56, 63, 70,
  8, 16, 24, 32, 40, 48, 56, 64, 72, 80,
  9, 18, 27, 36, 45, 54, 63, 72, 81, 90,
 10, 20, 30, 40, 50, 60, 70, 80, 90,100,
```

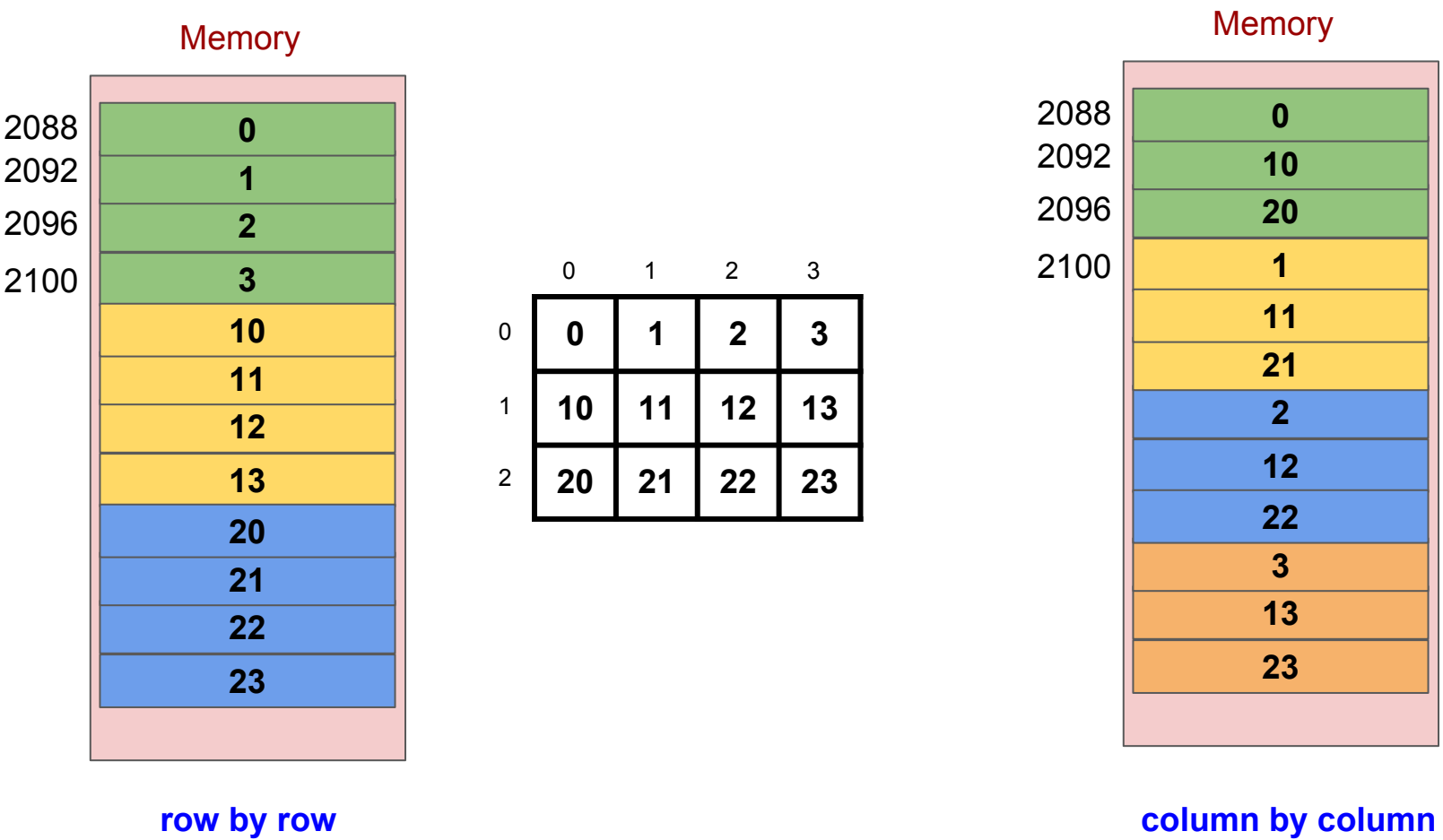
Remember: 1D arrays in memory

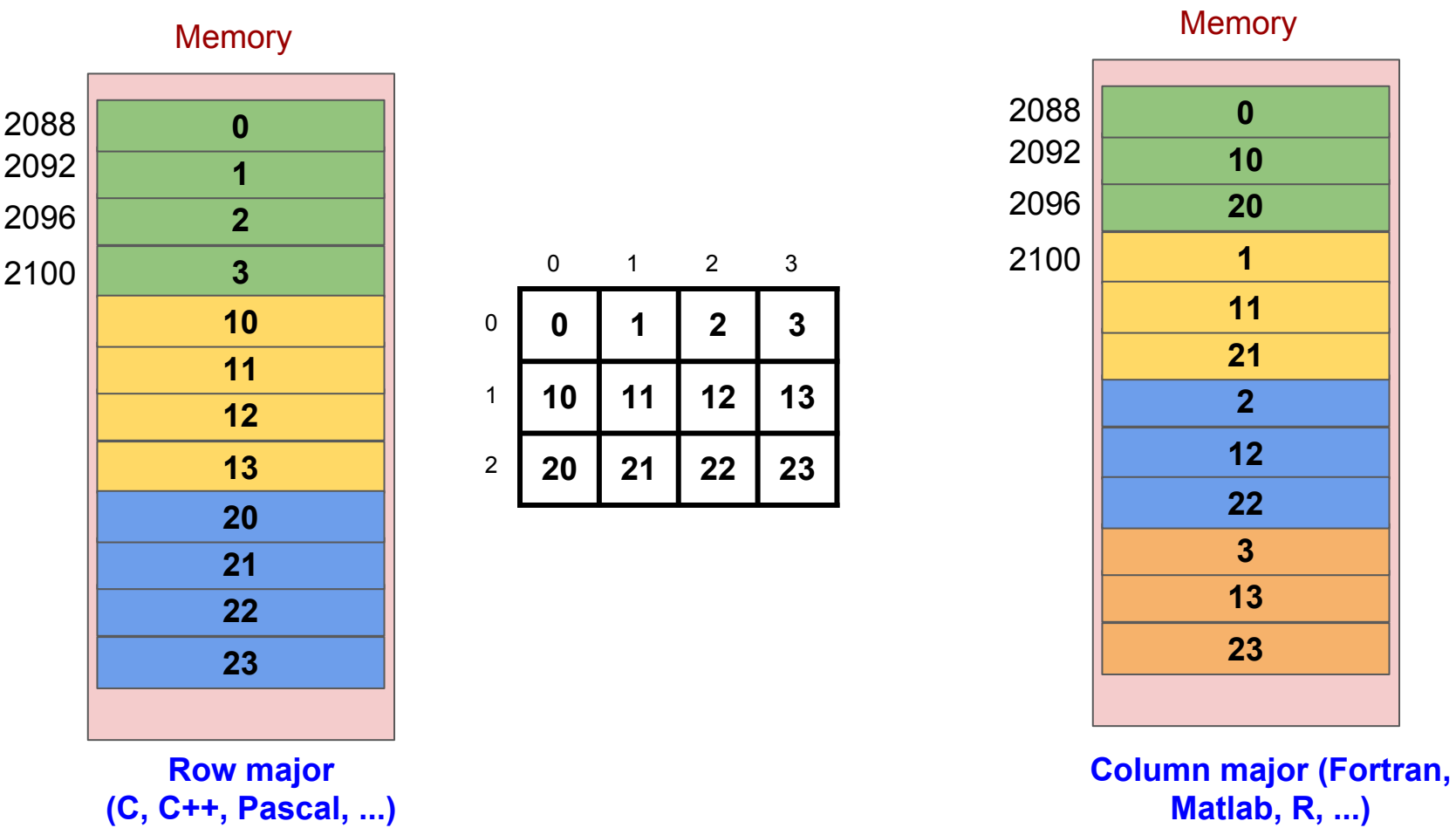
```
int a[7];
```

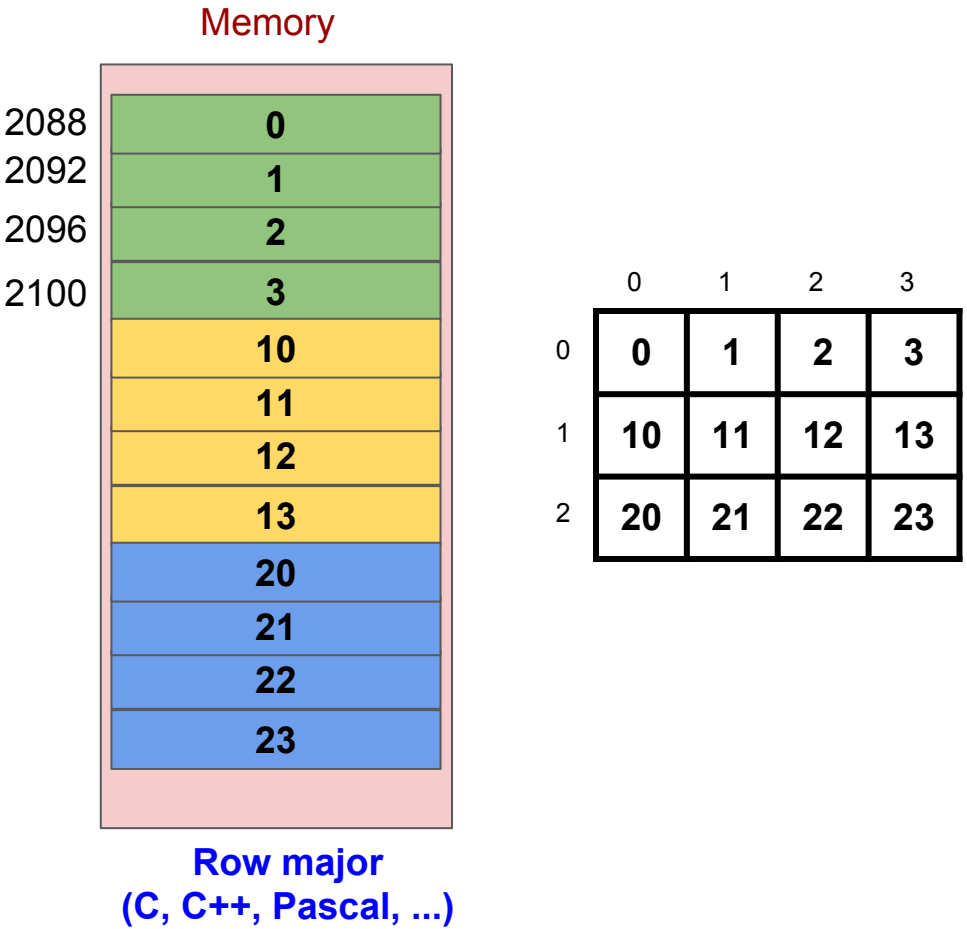


How are 2D arrays stored in memory?

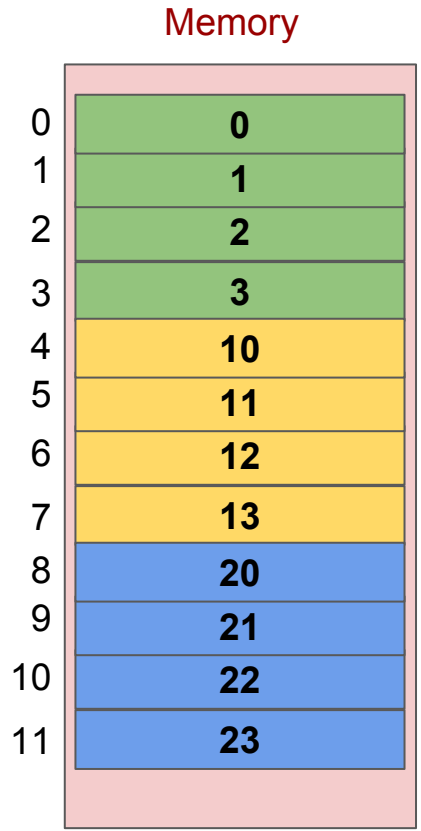
	0	1	2	3
0	**0**	**1**	**2**	**3**
1	**10**	**11**	**12**	**13**
2	**20**	**21**	**22**	**23**


Memory
2088
2092
2096
2100
0
1
2
3
10
11
12
13
20
21
22
23
row by row
0 1 2 3
0 0 1 2 3
1 10 11 12 13
2 20 21 22 23
Memory
2088
2092
2096
2100
0
10
20
1
11
21
2
12
22
3
13
23
column by column


Memory
2088
2092
2096
2100
0
1
2
3
10
11
12
13
20
21
22
23
Row major
(C, C++, Pascal, ...)
0
1
2
3
0
1
2
Memory
2088
2092
2096
2100
Column major (Fortran,
Matlab, R, ...)

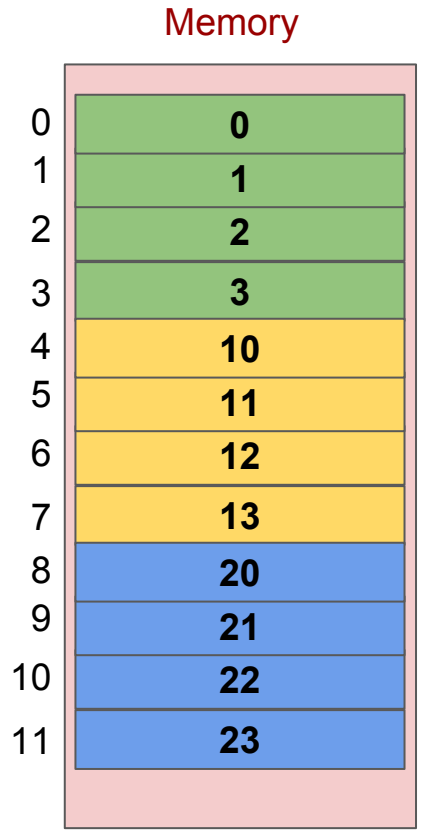


	0	1	2	3
0	0	1	2	3
1	10	11	12	13
2	20	21	22	23



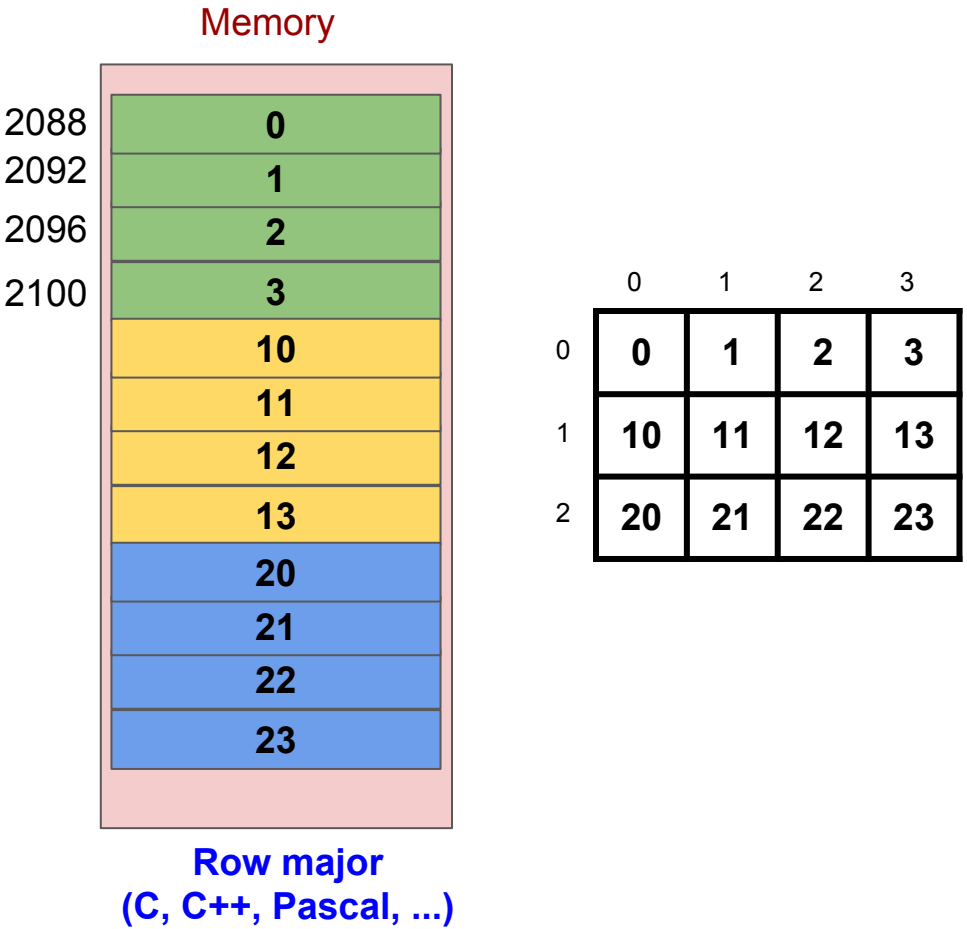
	0	1	2	3
0	0	1	2	3
1	10	11	12	13
2	20	21	22	23

```
a[0][1]
a[0][2]
a[0][j]

a[1][0]
a[1][1]
a[1][2]
a[1][j]

a[2][j]

a[i][j]
```

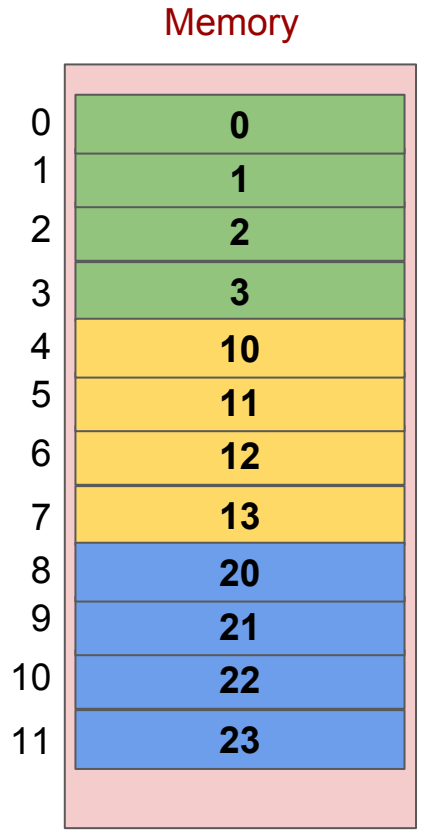


	0	1	2	3
0	0	1	2	3
1	10	11	12	13
2	20	21	22	23

```
a[0][1]
a[0][2]
a[0][j]

a[1][0]
a[1][1]
a[1][2]
a[1][j]

a[2][j]

a[i][j] : 4*i + j
```

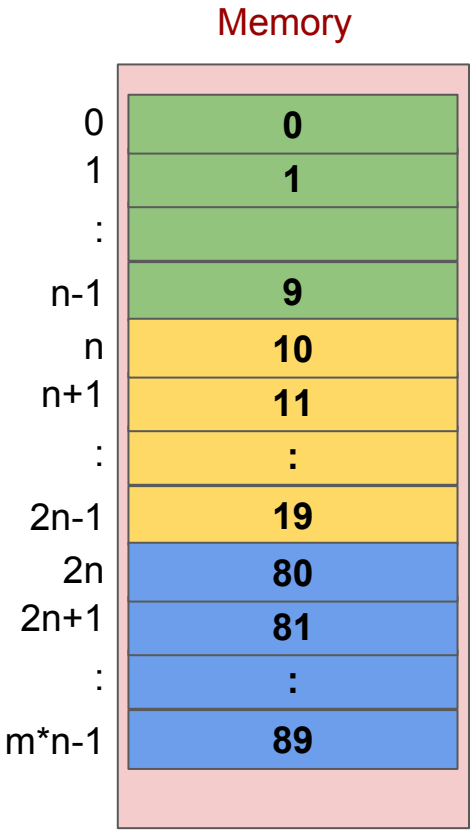


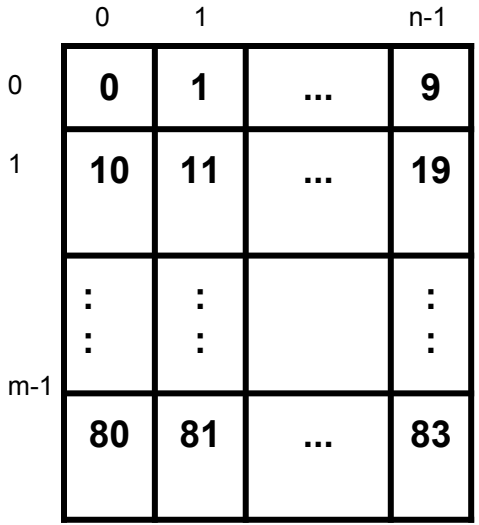
	0	1	2	3
0	0	1	2	3
1	10	11	12	13
2	20	21	22	23



	0	1	2	3
0	0	1	2	3
1	10	11	12	13
2	20	21	22	23

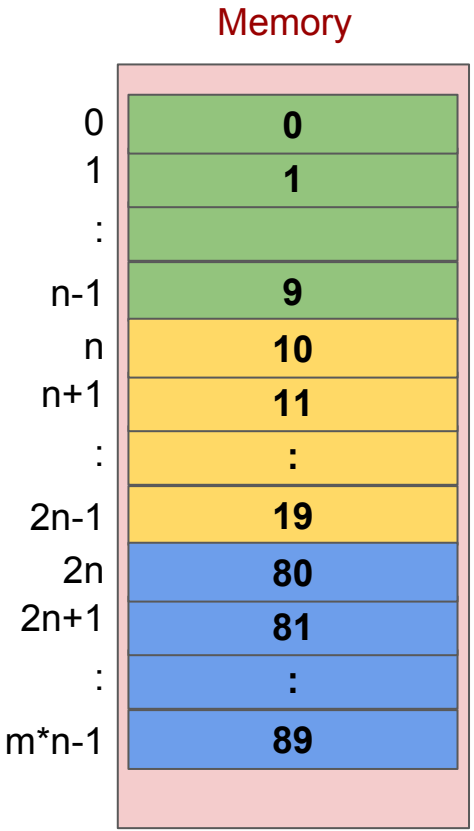
```
a[0][1]
a[0][2]
a[0][j]

a[1][0]
a[1][1]
a[1][2]
a[1][j]

a[2][j]

a[i][j]
```

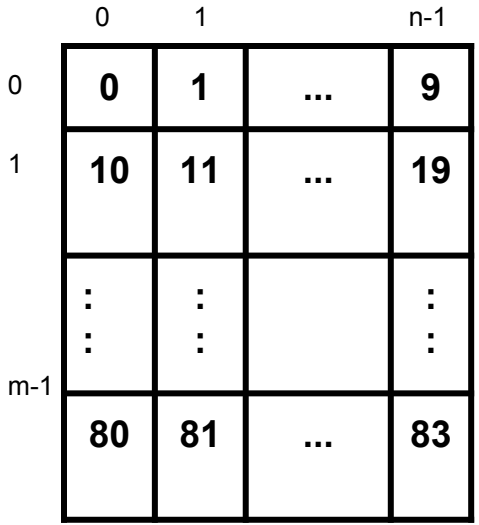





```
int a[m][n];
```

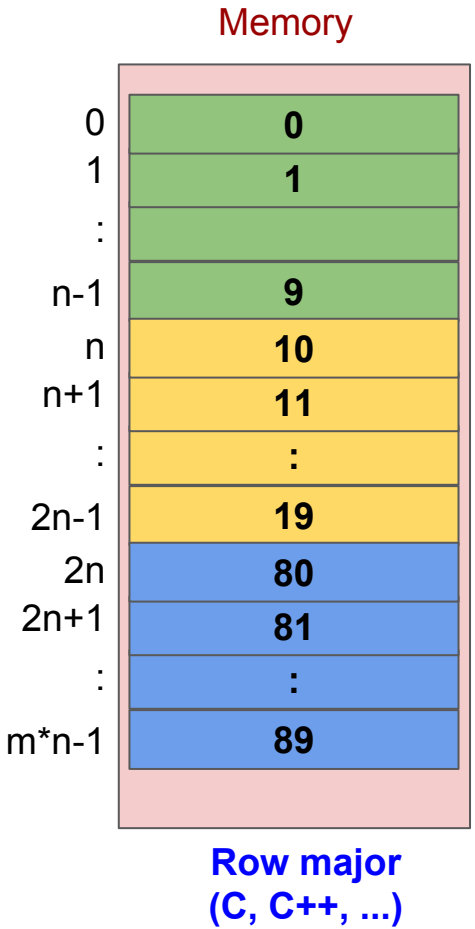
```
a[i][j] : n*i + j
```

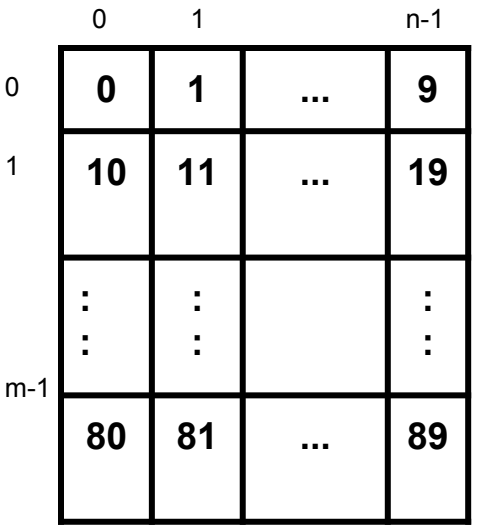





```
int a[m][n];
```

```
a[i][j] : n*i + j
```






```
int a[m][n];

a[i][j] : n*i + j
```

to find `a[i][j]` compiler needs to know:

- **m** (no. of rows of a)
- **n** (no. of columns of a)
- both m and n

Passing 2D arrays to functions

```c
#include <stdio.h>

int print2Darray(int a[][4], int,int);

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  print2Darray(a, 3,4);

  return 0;
}

int print2Darray(int a[][4], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }


}
```

print2darray1.c

Passing 2D arrays to functions

```c
#include <stdio.h>

#define N 4

int print2Darray(int a[][N], int,int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  print2Darray(a, 3,N);

  return 0;
}

int print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }
}
```

print2darray2.c

Accessing a row of 2D array

```c
#include <stdio.h>

#define N 4

void print2Darray(int a[][N], int,int);
void printArray(int a[], int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  printArray(a[1],4);

  return 0;
}

void print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}

void printArray(int arr[], int n) {
   for (int i = 0; i < n; i++)
      printf("%4d,", arr[i]);

   putchar('\n');
}
```

```c
#include <stdio.h>

#define N 4

void print2Darray(int a[][N], int,int);
void printArray(int a[], int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  printArray(a[1],7);

  return 0;
}

void print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}

void printArray(int arr[], int n) {
   for (int i = 0; i < n; i++)
      printf("%4d,", arr[i]);

   putchar('\n');
}
```

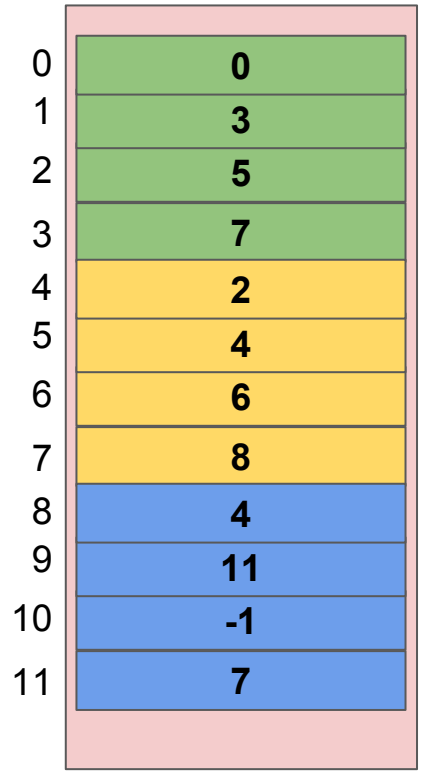
print2darray3.c

Accessing a row of 2D array

```c
#include <stdio.h>

#define N 4

void print2Darray(int a[][N], int,int);
void printArray(int a[], int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  printArray(a[1],4);

  return 0;
}

void print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}

void printArray(int arr[], int n) {
   for (int i = 0; i < n; i++)
      printf("%4d,", arr[i]);

   putchar('\n');
}
```

```c
#include <stdio.h>

#define N 4

void print2Darray(int a[][N], int,int);
void printArray(int a[], int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  printArray(a[1],7);

  return 0;
}

void print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}

void printArray(int arr[], int n) {
   for (int i = 0; i < n; i++)
      printf("%4d,", arr[i]);

   putchar('\n');
}
```

Main Memory



Row major
(C, C++, ...)

Passing 2D arrays to functions

```c
#include <stdio.h>

#define N 4

int print2Darray(int a[][N], int,int);

int main() {
  int a[3][N] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};


  print2Darray(a, 3,N);

  return 0;
}

int print2Darray(int a[][N], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }
}
```

print2darray2.c

Passing 2D arrays to functions

```c
#include <stdio.h>

void print2Darray(int m, int n, int a[][n]);

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  print2Darray(3,4,a);

  return 0;
}

void print2Darray(int m, int n, int a[][n]) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }
}
```

print2darray4.c

```c
#include <stdio.h>

void print2Darray(int a[][n], int m, int n);

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  print2Darray(3,4,a);

  return 0;
}

void print2Darray(int a[][n], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }
}
```

print2darray5.c

Passing 2D arrays to functions

```c
#include <stdio.h>

void print2Darray(int m, int n, int a[][n]);

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  print2Darray(3,4,a);

  return 0;
}

void print2Darray(int m, int n, int a[][n]) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}
```

```
   1,   3,   5,   7,
   2,   4,   6,   8,
   4,  11,  -1,   7,
```

print2darray4.c

```c
#include <stdio.h>

void print2Darray(int a[][n], int m, int n);

int main() {
  int a[3][4] = {{1, 3, 5, 7},
                 {2, 4, 6, 8},
                 {4,11,-1,7}};

  print2Darray(3,4,a);
```

```
print2Darray5.c: At top level:
print2Darray5.c:15:27: error: ‘n’ undeclared here (not in a function)
 void print2Darray(int a[][n], int m, int n) {
                           ^
```

```c
void print2Darray(int a[][n], int m, int n) {
   for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
      printf("%4d,", a[i][j]);

    putchar('\n');
   }

}
```