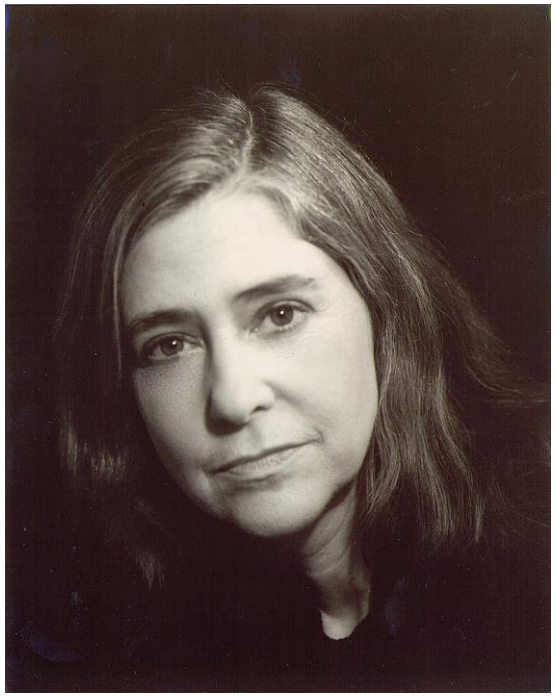
Two Dimensional Arrays

"What they used to do … was to assign you this program which nobody was able to ever figure out … It was tricky programming … comments were in Greek and Latin. So I was assigned this program and I actually got it to work. It even printed out its answers in Latin and Greek. I was the first one to get it to work."



By Daphne Weld Nichols, Photographer - Photograph of Margaret Hamilton taken by photographer Daphne Weld Nichols, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=12205173

Margaret Hamilton
Director of Apollo Flight Computer Programming

Based on slides by Chand John. Used with permission.

https://www.cs.utexas.edu/~chand/cs312/

Based on slides for Building Java Programs by Reges/Stepp, found at http://faculty.washington.edu/stepp/book/

2D Arrays in Java

- Arrays with multiple dimensions may be declared and used

```
int[][] mat = new int[3][4];
```

- The number of pairs of square brackets indicates the dimension of the array.
- By convention, in a 2D array the first number indicates the row and the second the column.

Two Dimensional Arrays

	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	**12**	0	0

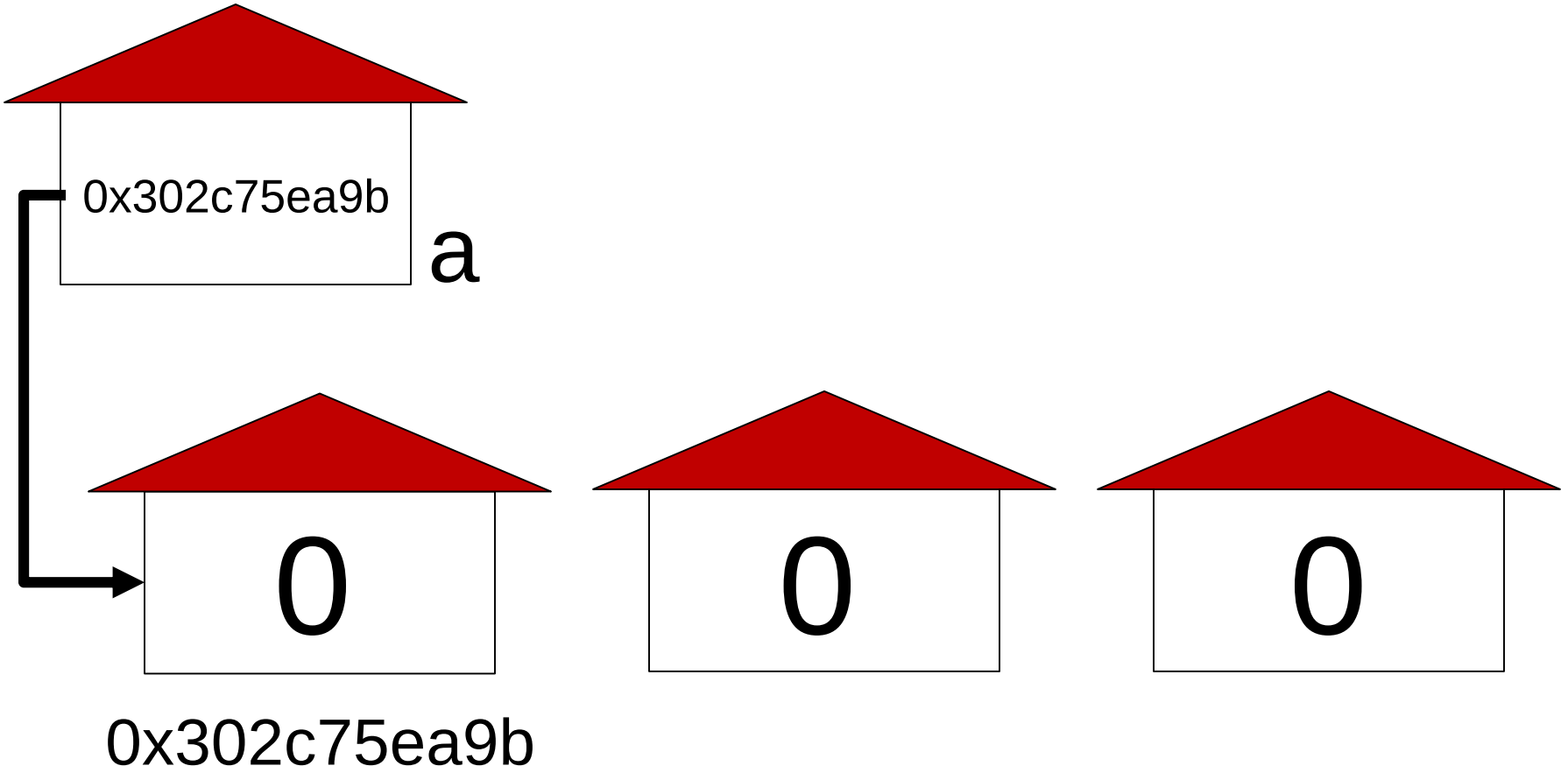
column

row

```
int[][] mat = new int[3][4];
mat[2][1] = 12;
```

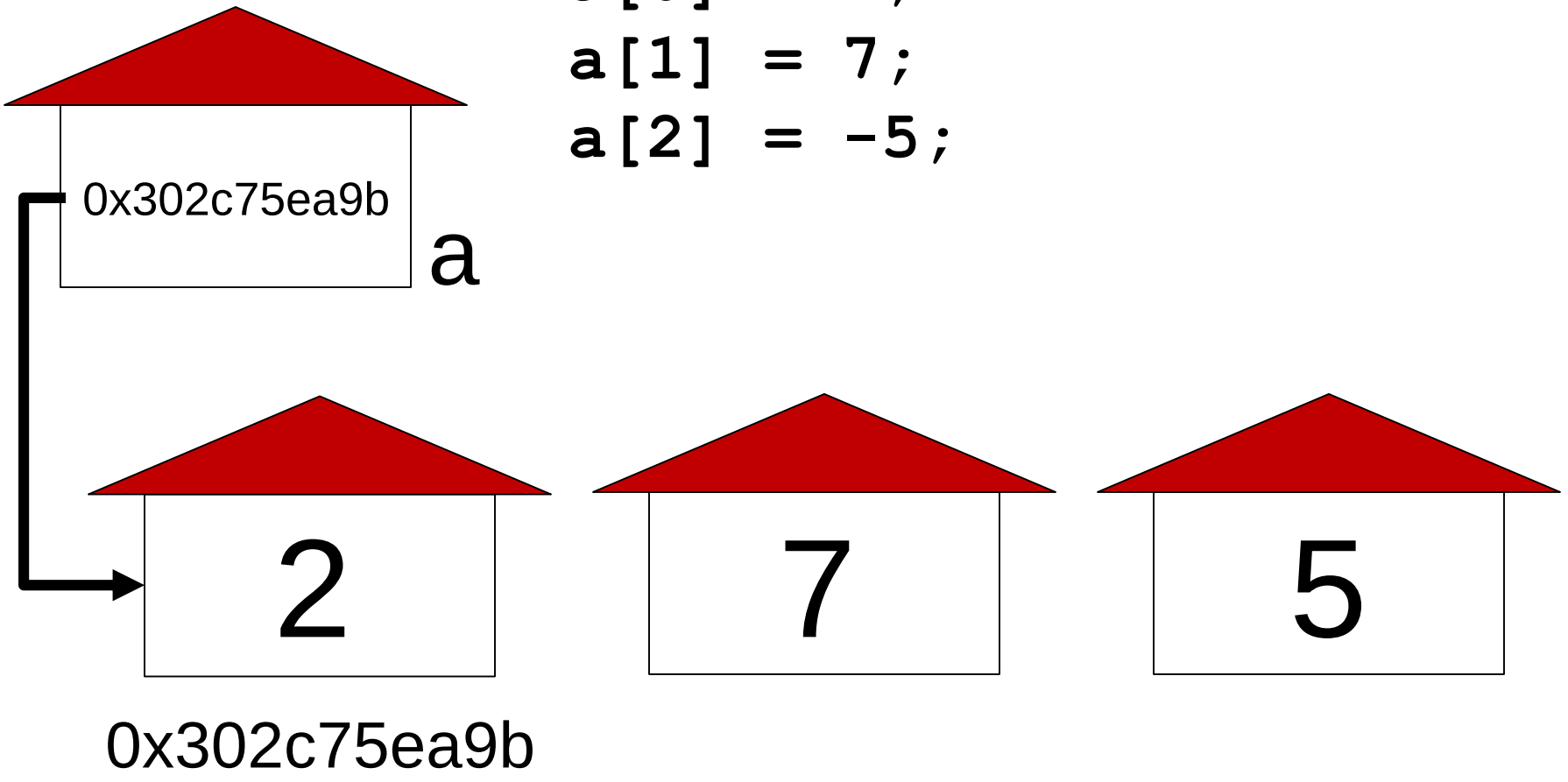
Remember how 1D arrays are stored in computer memory?

```
int[] a = new int[3];
```



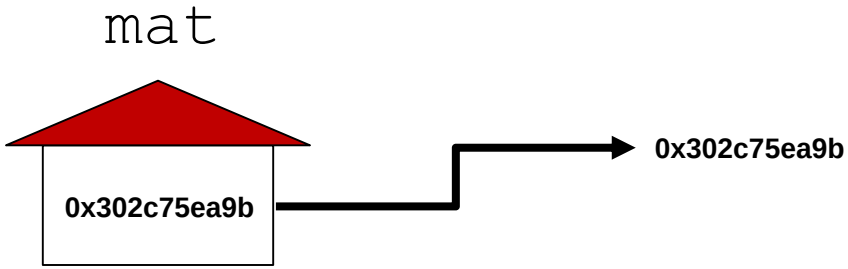
And how to access individual elements of a 1D array?

```
int[] a = new int[3];
a[0] = 2;
a[1] = 7;
a[2] = -5;
```



2D array = array of references

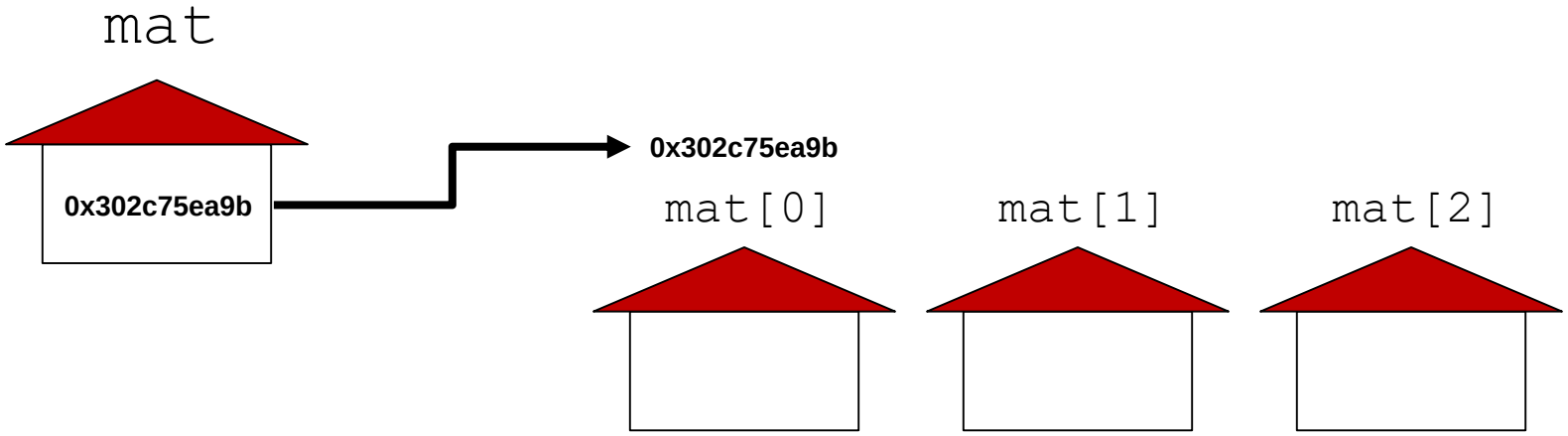
In the computer's memory:



`mat` is a reference (contains a memory address, i.e., "points to" a location in computer's memory…)

2D array = array of references

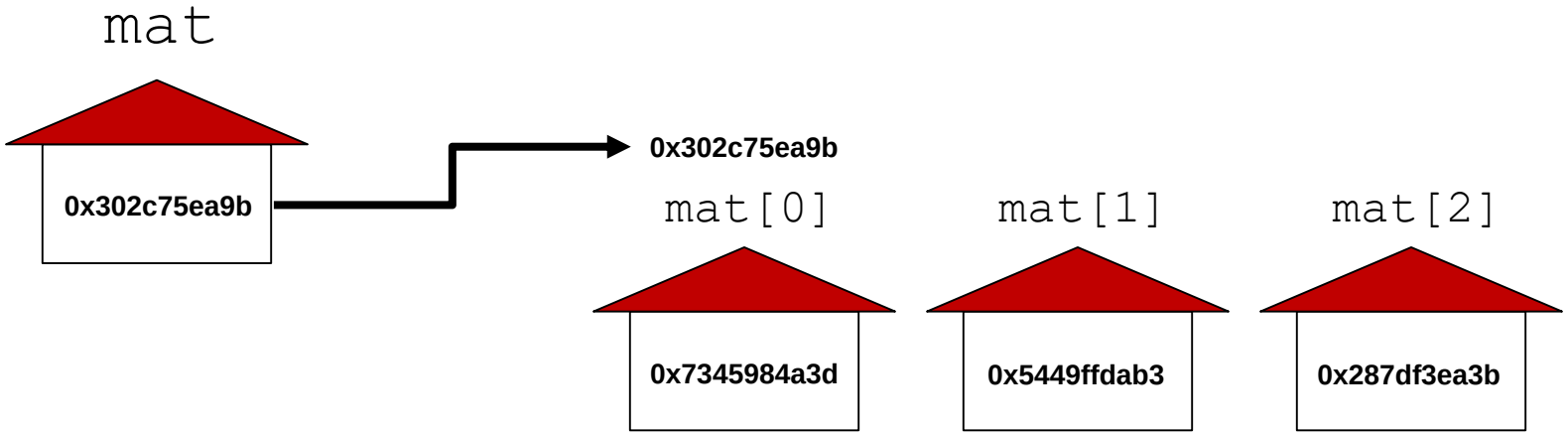
In the computer's memory:



At that memory location is `mat[0]`, followed by `mat[1]` right next to it, followed by `mat[2]`.

2D array = array of references

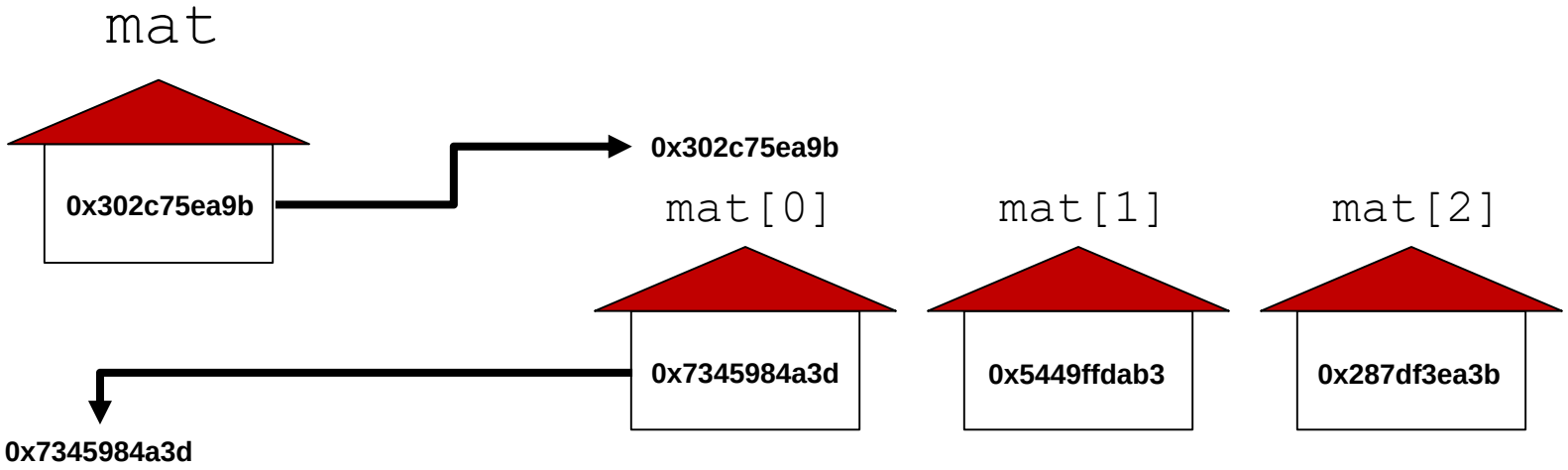
In the computer's memory:



Each of those array elements are themselves references to other memory locations…

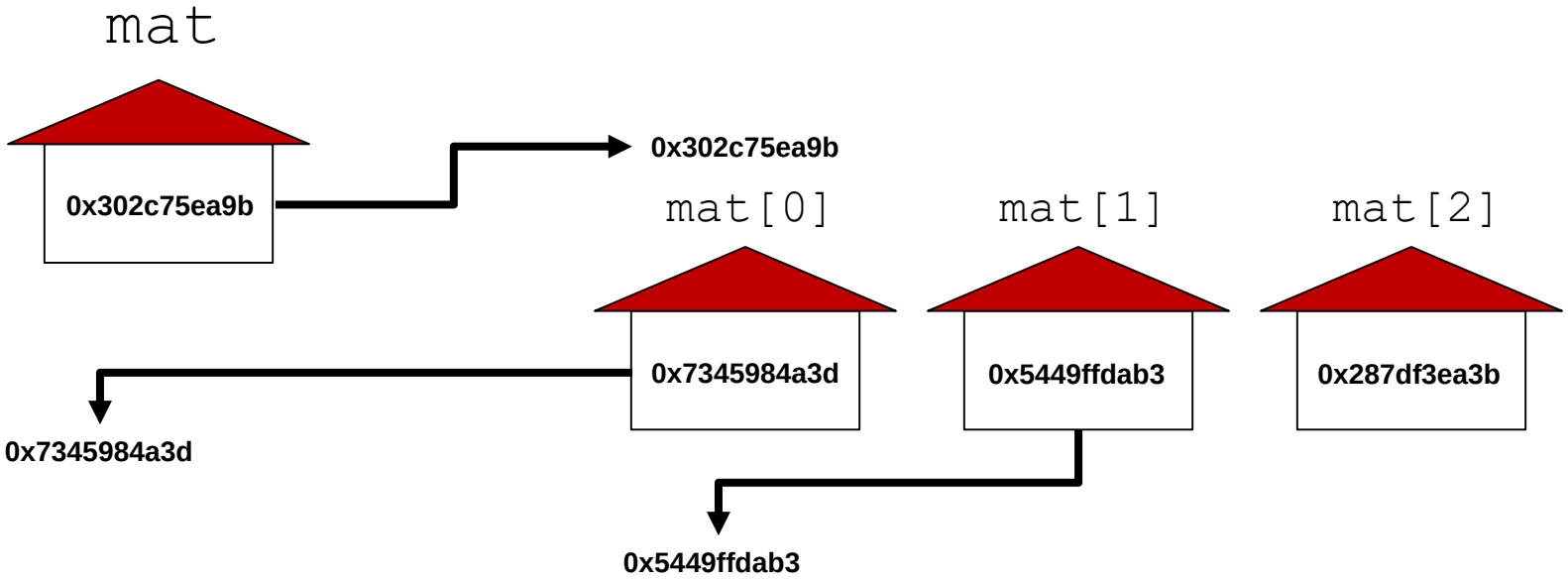
2D array = array of references

In the computer's memory:



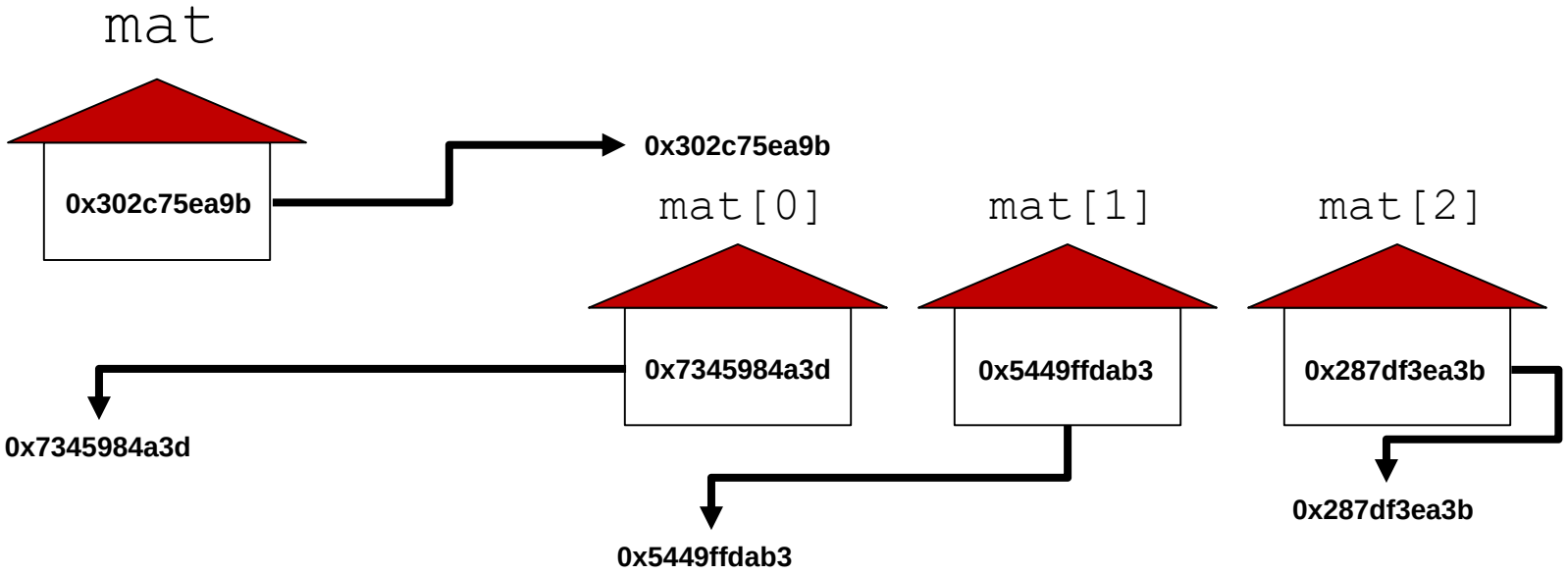
2D array = array of references

In the computer's memory:



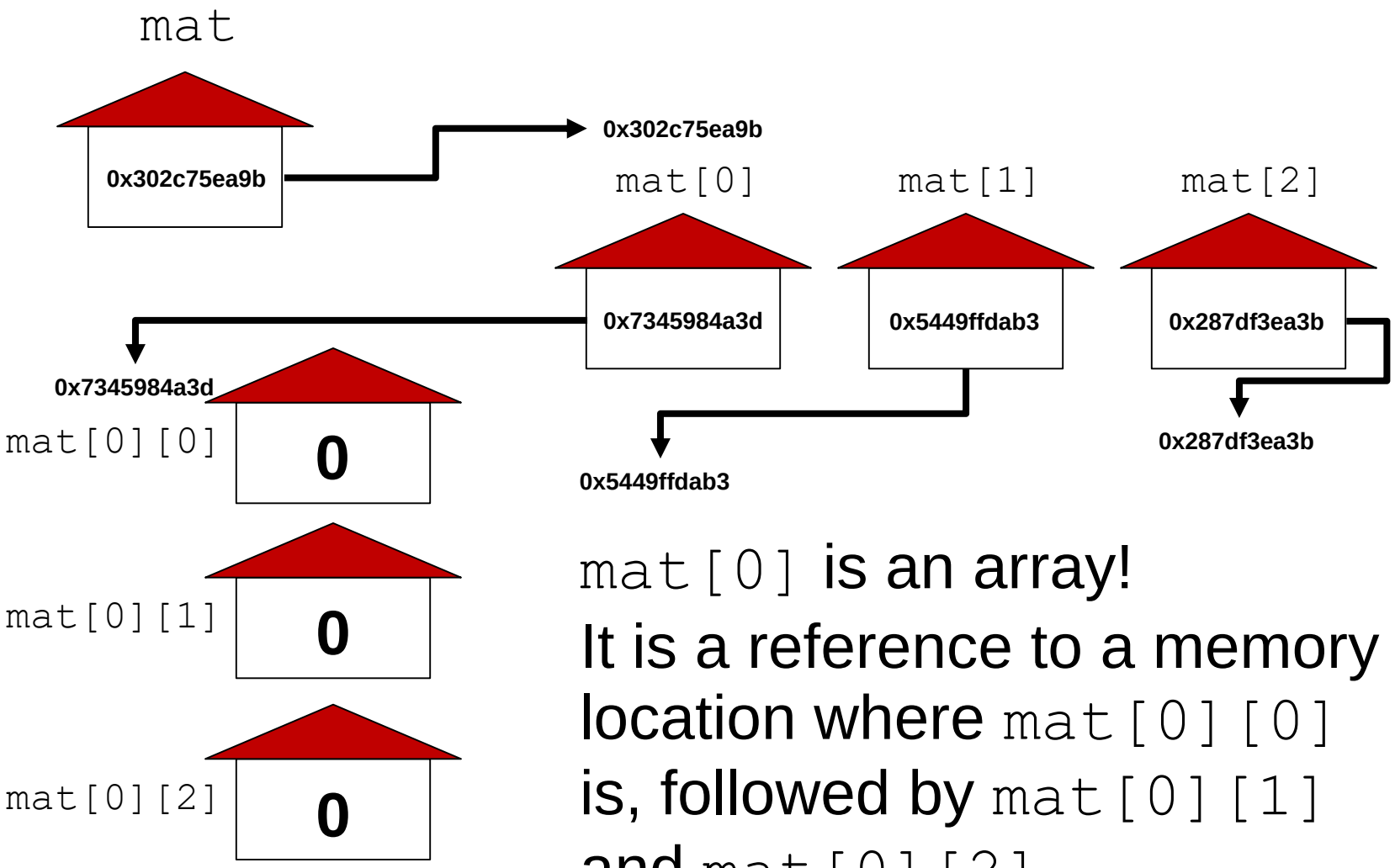
2D array = array of references

In the computer's memory:



2D array = array of references

In the computer's memory:

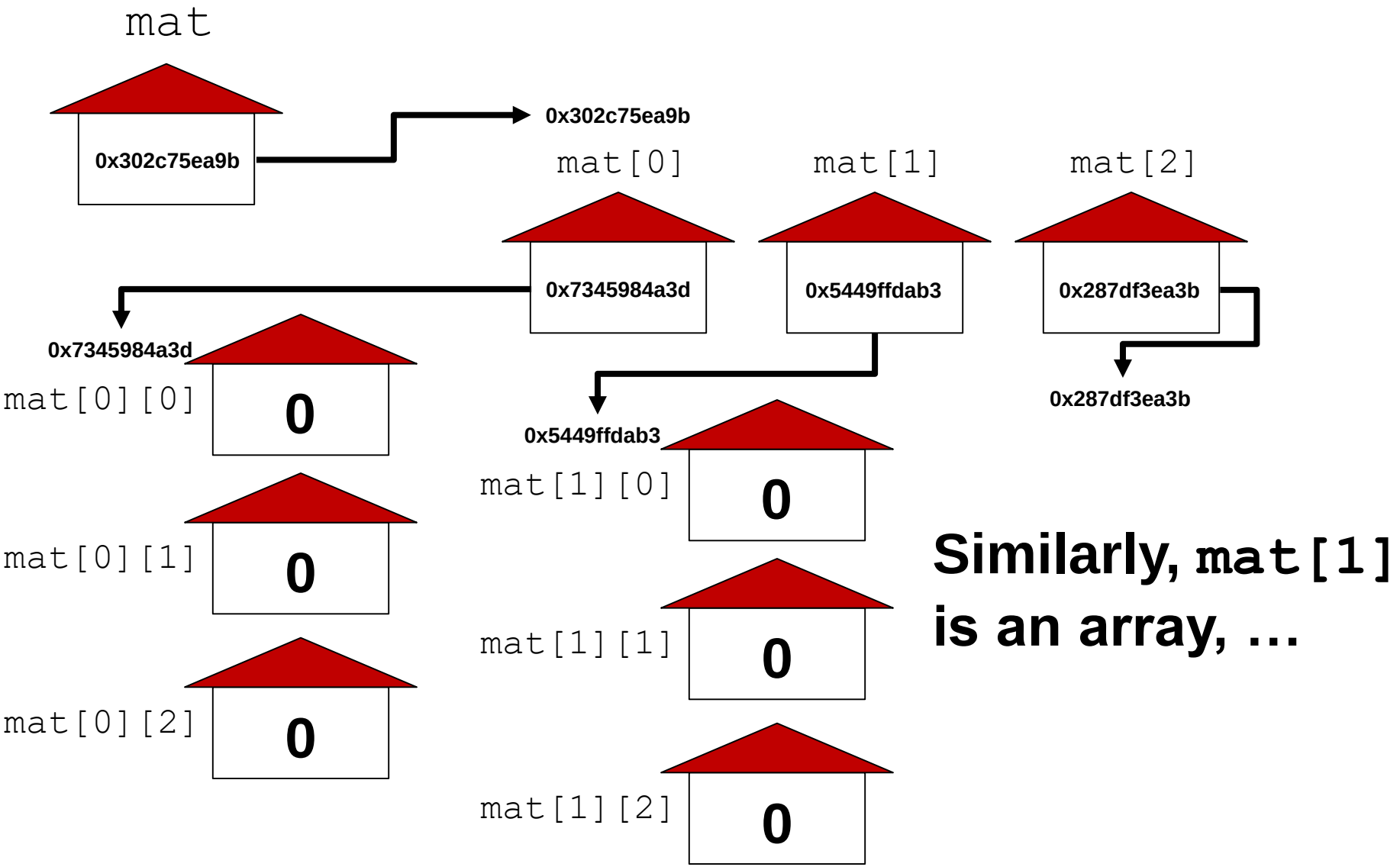


`mat[0]` is an array!

It is a reference to a memory location where `mat[0][0]` is, followed by `mat[0][1]` and `mat[0][2]`.

2D array = array of references

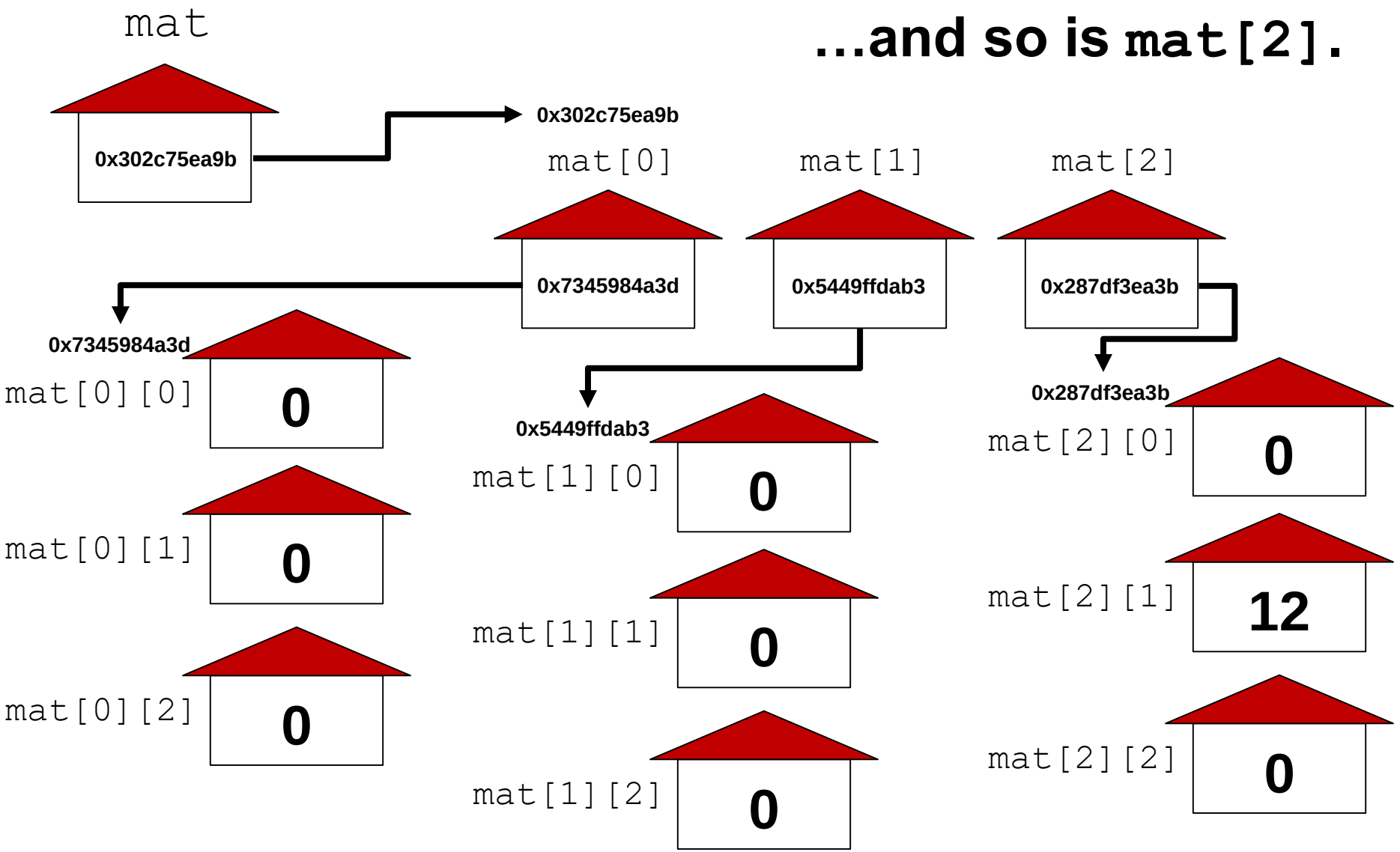
In the computer's memory:



Similarly, `mat[1]` is an array, …

2D array = array of references

In the computer's memory:

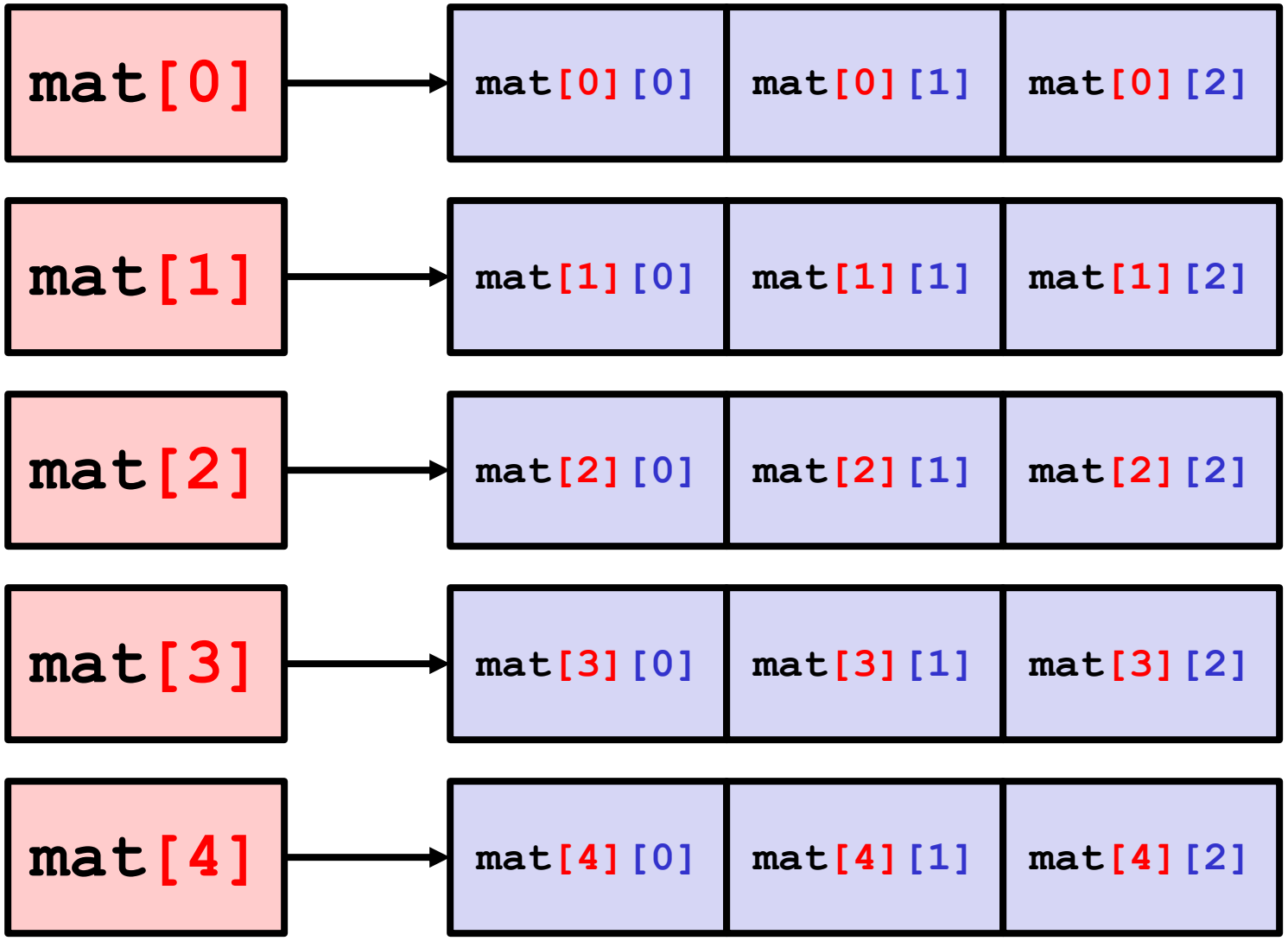


What is What?

```
int[][] mat = new int[10][12];

// mat is a reference to the whole 2d array

// mat[0] or mat[r] are references to a single row

// mat[0][1] or mat[r][c] are references to
// single elements

// no way to refer to a single column
```

Another picture of a 2D array

```
int[][] mat = new int[5][3];
```



2D Array Problems

- Write a method to find the maximum value in a 2D array of `int`s
- Write a method that finds the sum of values in each column of a 2D array of `double`s
- Write a method to print out the elements of a 2D array of `int`s in row order.
  - row 0, then row 1, then row 2 ...
- Write a method to print out the elements of a 2D array of `int`s in column order
  - column 0, then column 1, then column 2 ...

Exercise 1 strategy

- Go through entire 2D array and find maximum number. How??
- First, try a simpler problem: try to find the maximum value in a 1D array; just like you would do with a list of numbers on a sheet of paper:
  - Start at the first element of the array, and keep track of the biggest number you've seen so far.
  - If the second element is bigger, then set the biggest number you've seen so far to that.
  - If the third element is bigger than the biggest number you've seen so far, then set the biggest number you've seen so far to that.
  - Repeat until you've examined every element of the array.
  - Return the biggest value you've seen so far.
- A 2D array would work the same; you just need to "run through" the 2D array slightly differently since it has both rows and columns.

Exercise 1 - Possible Answer

```
public static int getMaxValueOf(int[][] numbers) {
    int max = numbers[0][0];

    for (int i = 0; i < numbers.length; i++) {
        for (int j = 0; j < numbers[i].length; j++) {
            if (numbers[i][j] > max) {
                max = numbers[i][j];
            }
        }
    }

    return max;
}
```

Full program using the method from Exercise 1

```
public class Cs312 {
    public static void main(String[] args) {
        int[][] numbers = {{39, 42, 1, 79, 2},
                           {41, 47, 7, 71, 9},
                           {38, 99, 2, 77, 8}};
        System.out.println("Maximum value is " + getMaxValueOf(numbers) + ".");
    }

    public static int getMaxValueOf(int[][] numbers) {
        int max = numbers[0][0];

        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                if (numbers[i][j] > max) {
                    max = numbers[i][j];
                }
            }
        }

        return max;
    }
}
```

Exercise 2 strategy

- What should the method return?
  - An array of column sums!
- Iterate over the columns of the 2D array.
  - For a single column, iterate through the rows of the column.
  - Calculate a cumulative sum of the values in the column → store the result in the column sum array.
- Return the column sum array.

Exercise 2 - Possible Answer

```
public static int[] getColumnSumsOf(int[][] numbers) {
     int numRows = numbers.length;
     int numColumns = numbers[0].length;
     int[] columnSums = new int[numColumns];

     for (int j = 0; j < numColumns; j++) {
        columnSums[j] = 0;
        for (int i = 0; i < numRows; i++) {
           columnSums[j] += numbers[i][j];
        }
     }

     return columnSums;
}
```

Full program using the method from Exercise 2

```
public class Cs312 {
    public static void main(String[] args) {
        int[][] numbers = {{39, 42, 1, 79, 2},
                           {41, 47, 7, 71, 9},
                           {38, 99, 2, 77, 8}};
        System.out.println("Column sums are: " + Arrays.toString(getColumnSumsOf(numbers)) +
".");
    }

    public static int[] getColumnSumsOf(int[][] numbers) {
        int numRows = numbers.length;
        int numColumns = numbers[0].length;
        int[] columnSums = new int[numColumns];

        for (int j = 0; j < numColumns; j++) {
            columnSums[j] = 0;
            for (int i = 0; i < numRows; i++) {
                 columnSums[j] += numbers[i][j];
            }
        }

        return columnSums;
    }
```

Exercise 3 - Possible Answer

```
public static void printInRowOrder(int[][] numbers) {
    int numRows = numbers.length;
    int numColumns = numbers[0].length;
    for (int i = 0; i < numRows; i++) {
        System.out.println(Arrays.toString(numbers[i]));
    }
}
```

Exercise 4 - Possible Answer

```
public static void printInColumnOrder(int[][] numbers) {
    int numRows = numbers.length;
    int numColumns = numbers[0].length;
    for (int j = 0; j < numColumns; j++) {
        System.out.print("Column #" + (j + 1) + ": [");
        System.out.print(numbers[0][j]);
        for (int i = 1; i < numRows; i++) {
            System.out.print(" ");
            System.out.print(numbers[i][j]);
        }
        System.out.println("]");
    }
}
```

clicker question

- What is output by the following code?

```
String[][] strTable = new String[5][8];
System.out.print(strTable.length + " ");
System.out.print(strTable[0].length + " ");
System.out.print(strTable[2][3].length());
```

A. 40 0 0

B. 8 5 0

C. 5 8 0

D. 5 8 then a runtime error occurs

E. No output due to a syntax error.

Use of Two Dimensional Arrays

- 2D arrays are often used when you need a table of data or want to represent things that have 2 dimensions.

- For example, a tic-tac-toe board…

Tic-tac-toe board as a 2D array

`board[0][0]`	`board[0][1]`	`board[0][2]`
`board[1][0]`	`board[1][1]`	`board[1][2]`
`board[2][0]`	`board[2][1]`	`board[2][2]`

Clicker Question

Write a program to play tic-tac-toe.

What data type do you want to use for the elements of the 2D array?

A. String
B. char
C. int
D. boolean
E. double

int[][] board = new int[3][3];

0 could mean “not filled”
1 could mean ‘X’
2 could mean ‘O’

`board[0][0]`	`board[0][1]`	`board[0][2]`
`board[1][0]`	`board[1][1]`	`board[1][2]`
`board[2][0]`	`board[2][1]`	`board[2][2]`

char[][] board = new char[3][3];

‘ ’ could mean “not filled”
‘X’ could mean ‘X’
‘O’ could mean ‘O’

`board[0][0]`	`board[0][1]`	`board[0][2]`
`board[1][0]`	`board[1][1]`	`board[1][2]`
`board[2][0]`	`board[2][1]`	`board[2][2]`