

Two-Dimensional Array

Contents of Lecture:

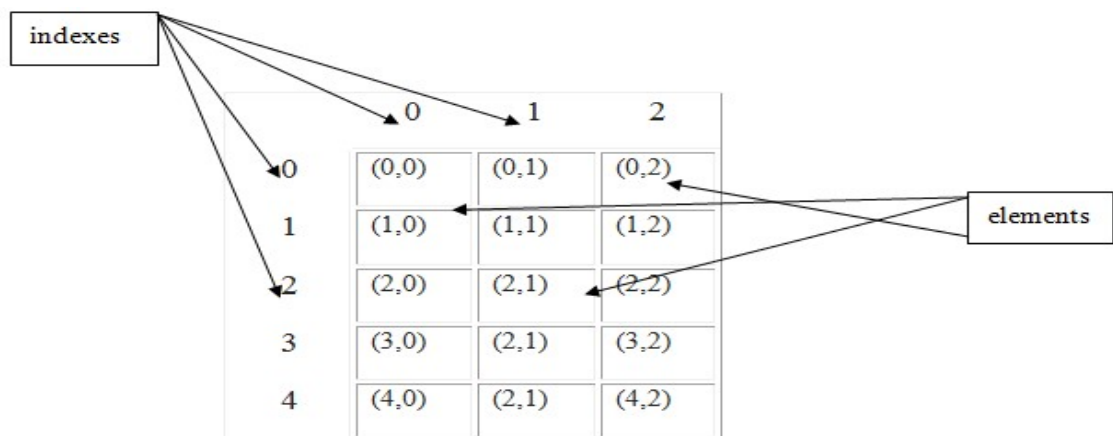
- ❖ Introduction
- ❖ Declaration and Initialization
- ❖ Processing 2D Arrays
- ❖ Example
- ❖ **Homework number (8) delivered next week**

Introduction

- ❖ While one-dimensional arrays allow data to be placed in an array one row at a time, two-dimensional arrays are capable of storing data in both rows and columns.
- ❖ To accomplish this, each row in a two-dimensional array is associated with the number of columns defined for the array.
- ❖ As with one-dimensional arrays, the entire array must contain elements of the same type.
- ❖ Because of the capability of storing data in rows and columns, it is obvious that two-dimensional arrays can provide more flexibility than one-dimensional arrays.

Declaration and Initialization

- ❖ Declaring a two-dimensional array has the following form:
Declare arr_name[*NUM_ROWS*][*NUM_COLS*]
- ❖ Where:
 - *arr_name* is the variable name for the array
 - *NUM_ROWS* is the maximum number of rows for the array
 - And *NUM_COLS* is the maximum number of columns for the array.
- ❖ To declare an array of five rows and three columns, the following code could be used:
 - Declare arr[5][3].



- Or declare a two-dimensional array by initial values; we can be accomplished using the following format:

```
int arr[5][3] =
{
  { 0, 1, 2 },
  { 3, 4, 5 },
  { 6, 7, 8 },
  { 9, 0, 1 },
  { 2, 3, 4 }
}
```

	0	1	2
0	0	1	2
1	3	4	5
2	6	7	8
3	9	0	1
4	2	3	4

The above code initializes:

```
arr[0][0] = 0
arr[0][1] = 1
arr[1][0] = 3
arr[1][1] = 4
arr[4][0] = 2
,...
```

Processing 2D Arrays

- ❖ Because 2D arrays must be filled by row and column, processing a 2D array can be done using nested for loops.
- ❖ For instance, to fill an array declared numArr[10][10] with user input, the following nested loop scheme could be used:

```
For row = 0 to 9
  For col = 0 to 9
    Prompt user for numArr[row][col]
    Get numArr[row][col]
    col ← col + 1
  EndLoop
  row ← row + 1
EndLoop
```

- ❖ To display the contents of the above filled array ten values per line, the following code could be used:

```

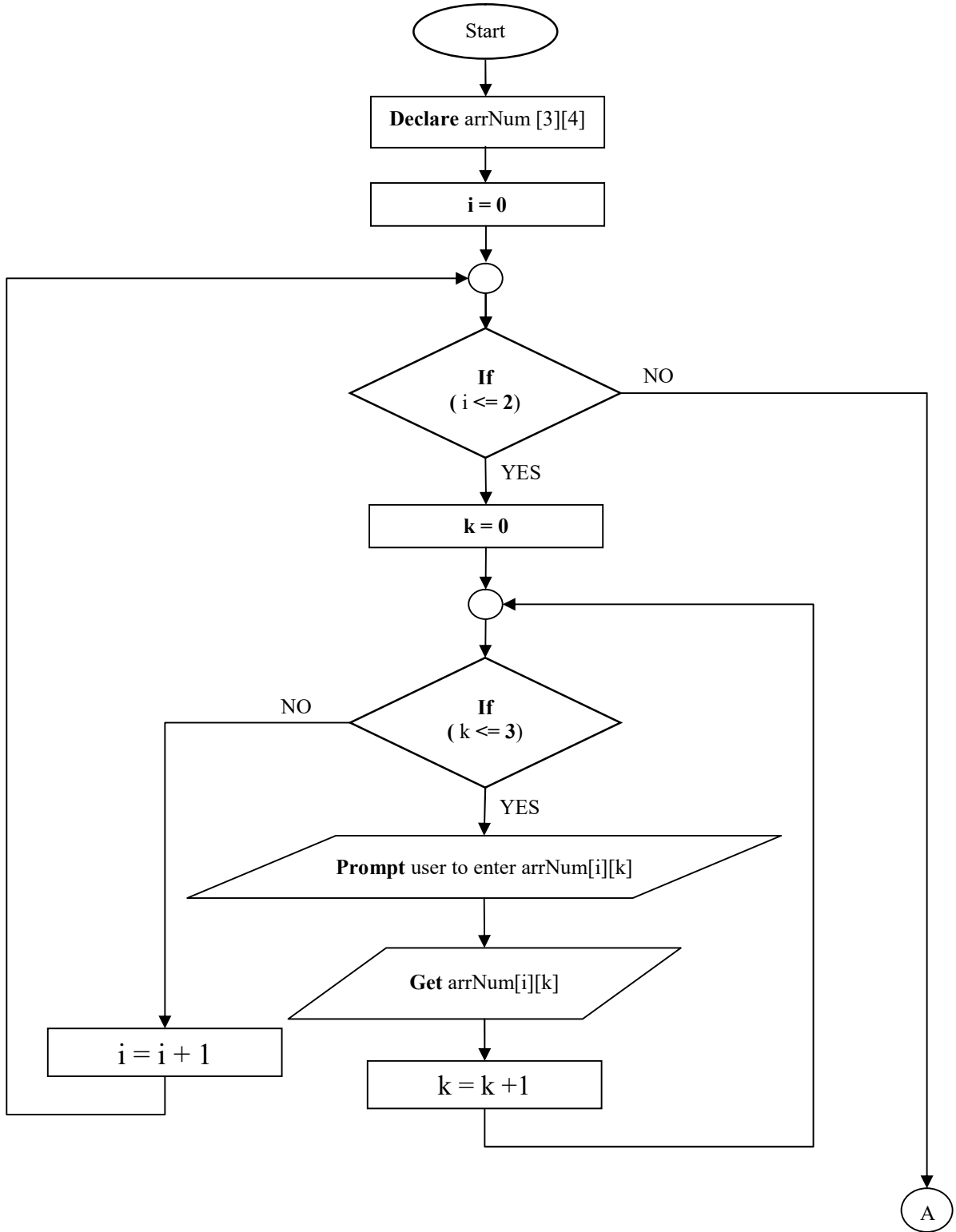
For row = 0 to 9
  For col = 0 to 9
    Display numArr[row][col]
    col ← col + 1
  EndLoop
  row ← row + 1
EndLoop

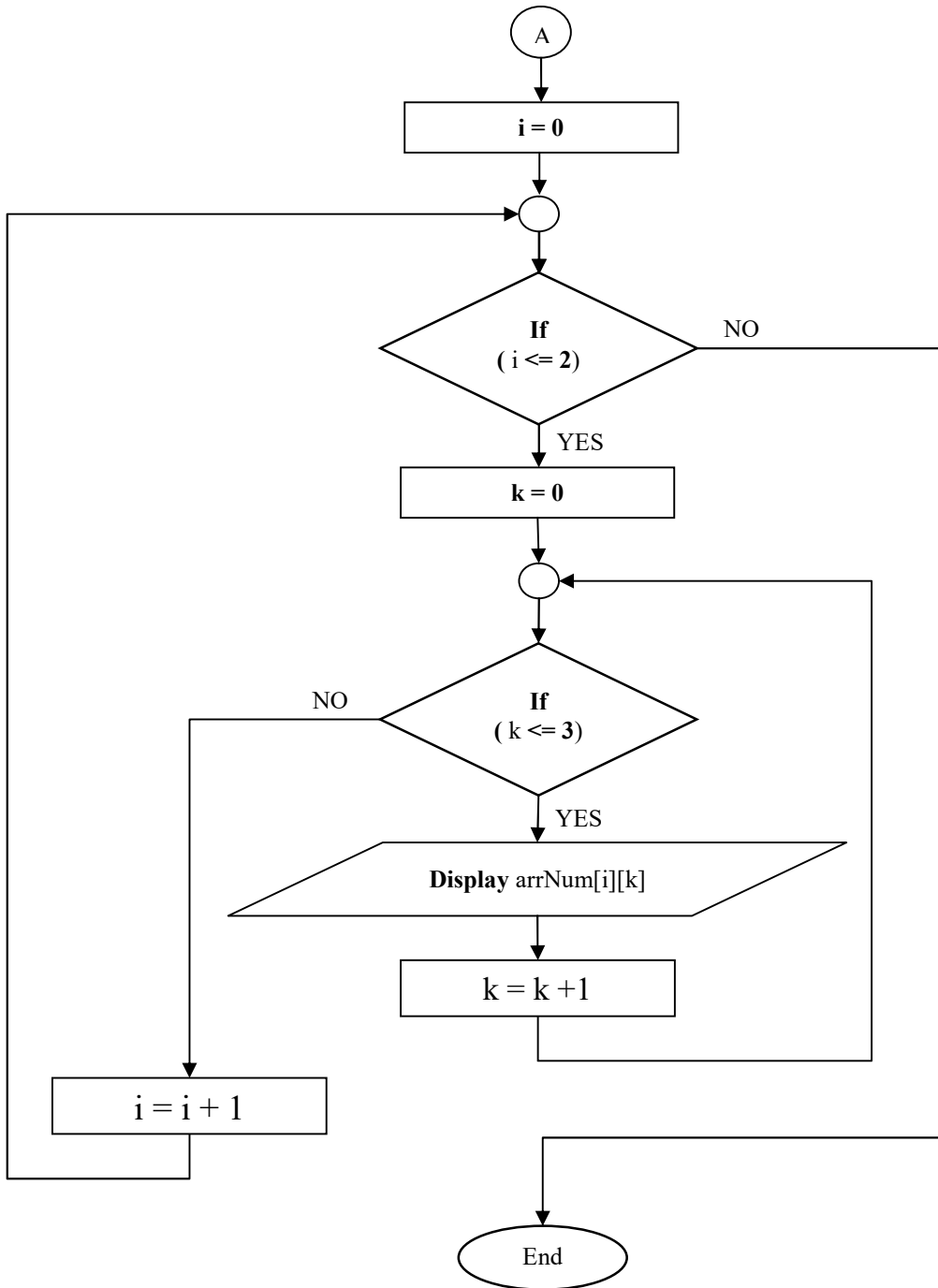
```

Example:

Write pseudo code and draw flowchart to input and output number in two-dimensional array of 3 rows and four column.

1. **Begin**
2. **Declare** arrNum[3][4]
3. **For** i = 0 to 3
 - For** k = 0 to 4
 - Prompt** user to enter arrNum[i][k]
 - Get** arrNum[i][k]
 - k = k + 1
 - EndLoop**
 - i = i + 1
 - EndLoop**
4. **For** i = 0 to 2
 - For** k = 0 to 3
 - Display** arrNum[i][k]
 - k = k + 1
 - EndLoop**
 - i = i + 1
 - EndLoop**
5. **End**





Homework number (8) delivered next week

- 1) Suppose we have a two dimensional number array of exactly ten rows and ten columns, and we need to find the sum of the integers along the main diagonal of the array. Write a pseudo code and draw a flowchart to do this operation.
- 2) Write a pseudo code and draw a flowchart that displays the maximum and minimal value in two-dimensional array that have size 6×4 .
- 3) write a pseudo code and draw a flowchart to display the summation of each row in two-dimension array of three rows and three columns, for example if you input values like the figure below

2	8	7
5	5	1
4	3	9

The output should like this:

17
11
16