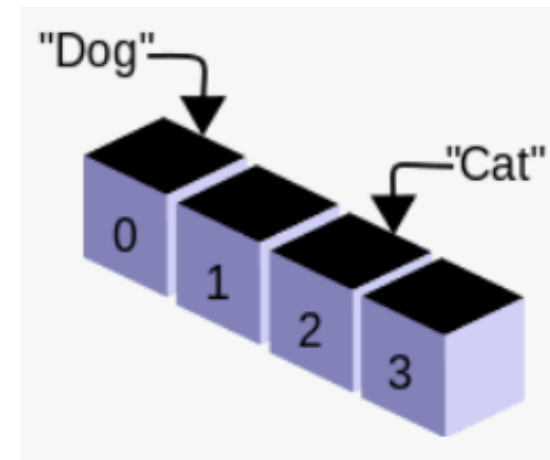


ALGORITHM & PROGRAMMING

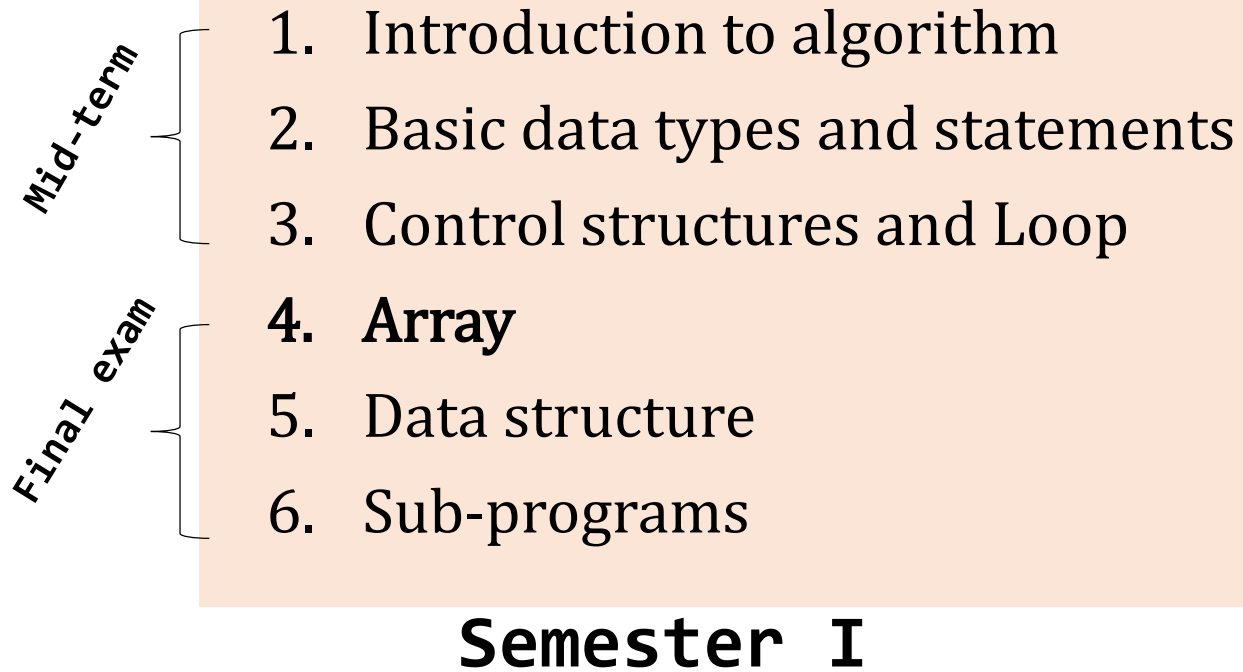
Chapter 4- Array

Prepared by: Mr. VALY Dona and Mr. NOU Sotheany
Modified and Lectured by: Mr. BOU Channa



Lecture overview

❑ Overall lectures



Lesson objectives

❑ Objectives

- Upon completion of this lesson, students will be able to
 - Understand a more advance type of data, called **Array**
 - Array of number
 - Array of string
 - Array of character ...
 - Know how to use different kind of array

Outline

❑ An overview of the lessons

■ Introduction

- Problem when not using array

■ Array

- What is array?
- How to use array?
- More on array

Introduction

❑ Problem

- **Problem #1:** Suppose we want to get 100 students' names then display their names in a list.

Will you use 100 variables?

```
Var name1, name2, ..., name100 : Sequence of characters
Begin
    read(name1, name2, ..., name100)
End
```

Disadvantages:

Too many creation of variables?
What if we have more than 100 variables?

- **Problem #2:** Suppose we want to get 100 subjects' scores of a students then do summation of those score.

Will we need 100 variables to store those scores?

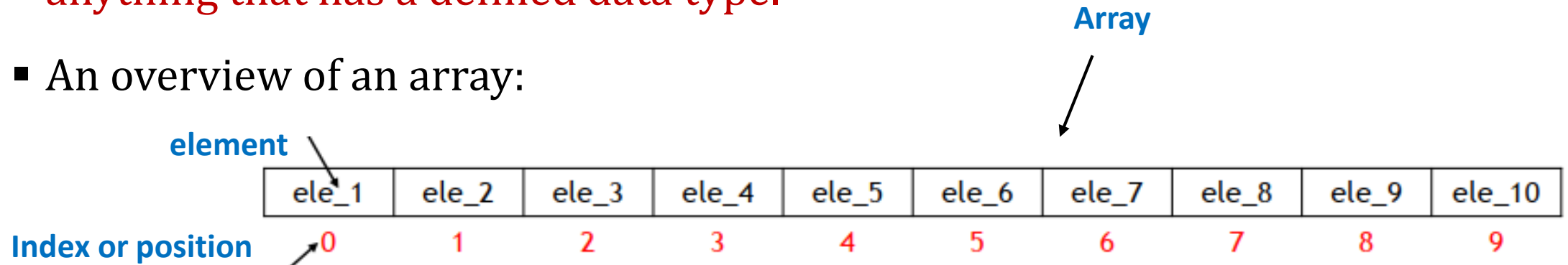
```
Var score1, score2, ..., score100, sum : float
Begin
    read(score1, score2, ..., score100)
    sum ← score 1+ score2 + ... + score100
End
```



Array

❑ What is an array?

- Array is a kind of data structure that stores many variables (elements) as a **single special variable**.
- Each **variable** in an array is called an **array element** and they have the **same variable type**
- You could have an **array of integers** or an **array of characters** or an array of anything that has a defined data type.
- An overview of an array:



Array

❑ Declaring (creating) an array

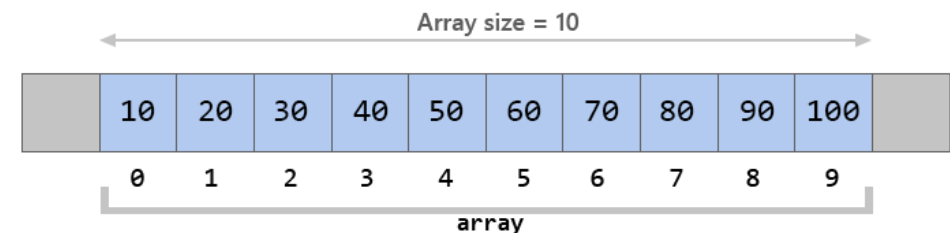
- To declare an array, we have to choose
 - Type of element in the array
 - Number of elements in the array
- Syntax

```
Var identifier[number of elements] : Type of element in array
```

- Examples: Creating array

```
Var num[20] : Integer
Var scores[10] : Float
Var name[50] : Array/sequence of characters
Var s[5][100] : Array of string (5 elements)
```

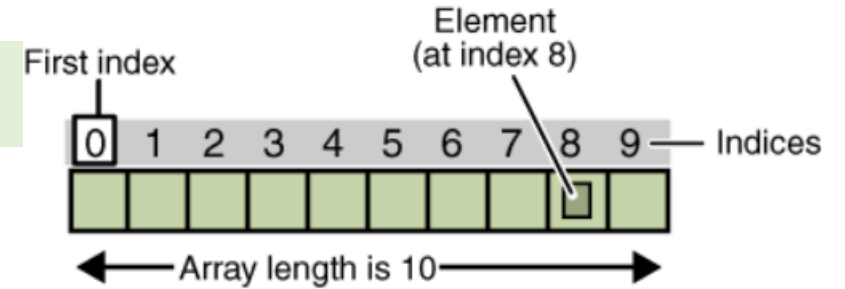
Suppose that we add values (10, 20, .., 100) to the array.
The array now look like this:



Array

□ Index or position

- In array, the value of index is
 - Start from 0 (some language may start with index 1)
 - E.g: In C language, index starts from 0 but in Matlab index starts from 1
 - Integer number
 - Last value of index is equal to number of elements in array minus 1 (when its index starts with 0)
- Index in the bracket can either be a direct integer value or a variable or an expression



```
Var score[10] : Integer
Begin
    score[0] ← 70
    score[1] ← 80
End
```

```
Var n : Integer
Var score[10] : Integer
Begin
    n ← 0
    score[n] ← 70
    score[n+1] ← 80
End
```


Array

❑ Access/use to an array

- To display array's elements, we need to access to each element
- To access a specific element in an array, use **arrayName[index]**
 - Ex: Suppose the array named **ele**
 - Then to access: **ele[0]**, **ele[1]**, ..., **ele[9]**
- Examples

```
Var scores[10] : Float
Var gender[50] : Sequence of characters
Begin
    read(scores[0], scores[1])
    read(gender[0], gender[1])
    write("Gender of student 1 is: ", gender, ".
He/she got ", scores[0], " scores.")
    write("Gender of student 2 is: ", gender[1], ".
He/she got ", scores[1], " scores.")
End
```

```
Var i : Integer
Var num[10] : Integer
Begin
    for (i←0; i<10; i←i+1) do
        read(num[i])
    end for

    for (i←0; i<10; i←i+1) do
        write(num[i])
    end for
End
```

What does this algorithm do?

Array

❑ Access/use to an array

- To access a specific element in an array, use `arrayName[index]`
 - Suppose we have an array named `ele`
 - Usage: `ele[0]`, `ele[1]`, ..., `ele[9]`

What does these algorithms do?

```
Var i : Integer
Var num[10] : Integer
Var s: Integer
Begin
    for (i←0; i<10; i←i+1) do
        read(num[i])
    end for
    s←0
    for (i←0; i<10; i←i+1) do
        s ← s + num[i]
    end for
    write(s)
End
```

- Get 10 numbers from the user.
- Then sum all those numbers together.
- Finally, display the result.

```
Var i : Integer
Var gender[10] : Sequence of character
Var m, n: Integer
Begin
    for (i←0; i<10; i←i+1) do
        read(gender[i])
    end for
    m←0
    n←0
    for (i←0; i<10; i←i+1) do
        if gender[i]=='M' then
            m++
        else if (gender[i]=='F' then
            n++
        end if
    end for
    write(m, n)
End
```

- Get 10 gender from the user.
- Then count all males and females
- Finally, display display #male, #female

Array

❑ Using array to solve the previous problems?

▪ Solution for Problem #1:

- *Use an array with the size of 100 and its type is a string (sequence of characters)*

```
Var names[100][20] : Sequence of characters
Begin
  for(i←0; i≤99; i++) do
    read(names[i])
  end for
End
```

▪ Solution for Problem #2:

- *Use an array with the size of 100 and its type is a float*
- Combine those variables into one by declaring an array then do loop to find summation.

```
Var scores[100] : float
Var sum : float
Begin
  sum ← 0
  for(i←0; i≤99; i++) do
    read(scores[i])
  end for

  for(i←0; i≤99; i++) do
    sum ← sum + scores[i]
  end for
  write("Total scores: ", sum)
End
```

Practice

□ Practice exercises

1. Write algorithms for the problems below :
 - a. Declare and store an array with 5 English's vowels
 - b. Declare and store an array with English's alphabet A-Z
 - c. Declare and store an array with even integer numbers 2, 4, ... 100
 - d. Declare and store an array of 10 user names. Ask the user to input all those 10 names. Then display their names on the screen
2. Write an algorithm to ask a user for 20 scores then
 - Find the average of those scores and show the scores that are greater than the average