Computer since

First Stage

Lec4

Array

MS.c Mortada Sabri

MS.c Esraa Hussei

Switch statement

A switch block conditionally executes one set of statements from several choices. Each choice is covered by a case statement.

The switch block tests each case until one of the cases is true. A case is true when

- For numbers, **eq(case_expression,switch_expression)**.
- For strings, **strcmp(case_expression,switch_expression)**.
- For objects that support the **eq(case_expression,switch_expression)**.
- For a cell array case_expression, at least one of the elements of the cell array matches switch_expression, as defined above for numbers, strings and objects.

When a case is true, MATLAB executes the corresponding statements and then exits the switch block.

The **otherwise** block is optional and executes only when no case is true.

Syntax

The syntax of switch statement in MATLAB is –

```
switch <switch_expression>
   case <case_expression>
      <statements>
   case <case_expression>
      <statements>
      ...
      ...
   otherwise
      <statements>
end
```

Example

Create a script file and type the following code in it –

```
grade = 'B';
  switch(grade)
  case 'A'
    fprintf('Excellent!\n' );
  case 'B'
    fprintf('Well done B\n' );
  case 'C'
    fprintf('Well done c\n' );
  case 'D'
    fprintf('You passed\n' );
  case 'F'
    fprintf('Better try again\n' );
  otherwise
    fprintf('Invalid grade\n' );
  end
```

What is an Array?

An array is a collection of similar data elements stored at contiguous memory locations. It is the simplest data structure where each data element can be accessed directly by only using its index number.

Need of using Array:

In programming, most of the cases need to store a large amount of data of similar type. To store such a huge amount of data, we need to define numerous variables. It would be very tough to memorize all variable names while writing the programs. Instead, it is better to define an array and store all the elements into it.

Advantages of Array:

1. Arrays represent multiple data elements of the same type using a single name.
2. In an array, accessing or searching an element is easy by using the index number.
3. An array can be traversed easily just by incrementing the index by 1.
4. Arrays allocate memory in contiguous memory locations for all its data elements.

Accessing Elements in an Array:

To access any element of an array we need the following details:

1. Base Address of array.
2. Size of an element in bytes.
3. Which type of indexing, array follows.

Creating Vectors:

A vector is a one-dimensional array of numbers. MATLAB allows creating two types of vectors:

Row vectors

Column vectors

Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.

For example,

r = [7 8 9 10 11]

MATLAB will execute the above statement and return the following result:

```
r =

Columns 1 through 4

7 8 9 10

Column 5

11
```

Another example,

```
r = [7 8 9 10 11];

t = [2, 3, 4, 5, 6];

res = r + t
```

MATLAB will execute the above statement and return the following result:

```
res =

Columns 1 through 4

9 11 13 15

Column 5

17
```

Column vectors are created by enclosing the set of elements in square brackets, using semicolon (;) to delimit the elements.

```
c = [7; 8; 9; 10; 11]
```

MATLAB will execute the above statement and return the following result:

```
C =

7

8
```

```
9

10

11
```

Creating Matrices:

A matrix is a two-dimensional array of numbers.

In MATLAB, a matrix is created by entering each row as a sequence of space or comma separated elements, and end of a row is demarcated by a semicolon. For example, let us create a 3-by-3 matrix as:

```
m = [1 2 3; 4 5 6; 7 8 9]
```

MATLAB will execute the above statement and return the following result:

```
m =

1 2 3

4 5 6

7 8 9
```