```
A =
     2     5     5    10    15    20
     3     6     9    12    15    18
     4     7    30    35    40    45
     5     8    95    94    93    92
     6     9    60    65    70    75
```

2.10 *Strings and Strings as Variables*

- A string is an array of characters. It is created by typing the characters within single quotes.
- Strings can include letters, digits, other symbols, and spaces.
- Examples of strings: `'ad ef '`, `'3%fr2'`, `'{edcba:21!'`, `'MATLAB'`.
- A string that contains a single quote is created by typing two single quotes within the string.
- When a string is being typed in, the color of the text on the screen changes to maroon when the first single quote is typed. When the single quote at the end of the string is typed, the color of the string changes to purple.

Strings have several different uses in MATLAB. They are used in output commands to display text messages (Chapter 4), in formatting commands of plots (Chapter 5), and as input arguments of some functions (Chapter 7). More details are given in these chapters when strings are used for these purposes.

- When strings are being used in formatting plots (labels to axes, title, and text notes), characters within the string can be formatted to have a specified font, size, position (uppercase, lowercase), color, etc. See Chapter 5 for details.

Strings can also be assigned to variables by simply typing the string on the right side of the assignment operator, as shown in the examples below:

```
>> a='FRty 8'
a =
FRty 8
>> B='My name is John Smith'
B =
My name is John Smith
>>
```

When a variable is defined as a string, the characters of the string are stored in an array just as numbers are. Each character, including a space, is an element in the array. This means that a one-line string is a row vector in which the number of elements is equal to the number of characters. The elements of the vectors are addressed by position. For example, in the vector `B` that was defined above the 4th element is the letter n, the 12th element is J, and so on.

```
>> B(4)
ans =
n
>> B(12)
ans =
J
```

As with a vector that contains numbers, it is also possible to change specific elements by addressing them directly. For example, in the vector B above the name John can be changed to Bill by:

```
>> B(12:15)='Bill'
B =
My name is Bill Smith
>>
```

Using a colon to assign new characters to elements 12 through 15 in the vector B.

Strings can also be placed in a matrix. As with numbers, this is done by typing a semicolon ; (or pressing the **Enter** key) at the end of each row. Each row must be typed as a string, which means that it must be enclosed in single quotes. In addition, as with a numerical matrix, all rows must have the same number of elements. This requirement can cause problems when the intention is to create rows with specific wording. Rows can be made to have the same number of elements by adding spaces.

MATLAB has a built-in function named char that creates an array with rows having the same number of characters from an input of rows not all of the same length. MATLAB makes the length of all the rows equal to that of the longest row by adding spaces at the end of the short lines. In the char function, the rows are entered as strings separated by a comma according to the following format:

```
variable_name = char('string 1','string 2','string 3')
```

For example:

```
>> Info=char('Student Name:','John Smith','Grade:','A+')
Info =
Student Name:
John Smith
Grade:
A+
>>
```

A variable named Info is assigned four rows of strings, each with different length.

The function char creates an array with four rows with the same length as the longest row by adding empty spaces to the shorter lines.