CSE331L_2 - Variables, I/O ; Array.

Creating Variables:

Syntax for a variable declaration:

name DB value
name DW value

DB = stands for Define Byte.

DW = stands for Define Word.

- name - can be any letter or digit combination though it should start with a letter. It's possible to declare unamed variables by not specifying the name.
- value : can be any numeric value in any supported numbering system or "?" symbol for variables that are not initialized.

Creating Constants

Constants are just like variables, but they exist only until your program is compiled. After definition of a constant its value cannot be changed. To define constants EQU directive is used:

name EQU <arg expressions>

For example:

```
K EQU 5
MOV AX, K
```

Creating Arrays

Arrays can be seen as chains of variables. A text string is an example of a byte array, each character is presented as an ASCII code value.

```
a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
b DB 'Hello', 0
```

- You can access the value of any element in array using square brackets, for example:

```
MOV AL, a[3]
```

- You can also use any of the memory index registers BX, SI, DI, BP, for example:

```
MOV SI, 3
MOV AL, a[SI]
```

- If you need to declare a large array you can use DUP operator.

The syntax for DUP:

```
c DB 5 DUP (9)
```

is an alternative way of declaring:

```
c DB 9,9,9,9,9
```

Memory Access

To access memory, we can use these ~~four use for~~ four registers: BX, SI, DI, BP. Combining these registers inside [] symbols, we can get different memory locations.