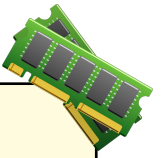

3	.	1	4	1	5	□

id:	63
kind:	DBL_VAL
data.int_val:	
data.dbl_val:	3.1415□
data.txt_val:	

10.0
-5
17.21
25.1

id:	2
kind:	TXT_VAL
data.int_val:	
data.dbl_val:	
data.txt_val:	Fred□

INT_VAL

73

Func or Proc

val:	5
ptr:	0002
Instruction:	Step 2

```
Function: Read Row
------------------------------
Returns: Row - a Row with data read from the user
Parameters:
  1: next id (Integer) - the id of the row to be read
Local Variables:
  *: line (String - 16 characters) - the text read from the user
Steps:
  1: Set result's id to next id
  2: Output 'Enter value: ' to the Terminal
  3: Read text entered by user into line
  4: If line is an integer
  5:     set result's data's Int Val to the integer value in line
  6:     set result's kind to INT_VAL
  7: Else If line is a double
  8:     set result's data's Dbl Val to the double value in line
  9:     set result's kind to DBL_VAL
 10: Else
 11:     set result's data's Txt Val to the text in line
 12:     set result's kind to TXT_VAL
 13: Output "Stored in row with id ", and result's id
 14: Return the result

Procedure: Main
------------------------------
Local Variables:
  *: db_data (array containing 3 Row values)
  *: i (Integer) -
Steps:
  1: for i loops over each element in db_data
  2:     set db_data[i] to result of calling Read Row(i)
  3: ...
```

A pointer can be used to "point" to the location of the data on the heap

The pointer's value can be stored in a local variable on the stack... and used to access the value(s) on the heap