Quarter 2 Project – Text Excel Part A

Advanced Placement Computer Science

Bellevue High School

Contents

Introduction

The very first "killer app" for the personal computer that made it essential for business was VisiCalc[1]. It was a spreadsheet program that showed the number crunching prowess and automation capabilities of the personal computer. It moved the computer out of the hobbyist's home and into corporate America.

The objective of this assignment will be to create a lightweight version of the spreadsheet program on the console window.

Overview

A spreadsheet is a two dimensional series of cells indexed by column letters and row numbers. A letter followed by a number identifies cells within spreadsheets (e.g., "D13" or "F9"). The cell

[1] Available: http://www.bricklin.com/history/vcexecutable.htm

designated by “C2” means the third column (labeled “C”) and the 2nd row, as in this partial printout of a spreadsheet:

```
    |A          |B          |C          |D
1   |           |           |           |
2   |           |           |(cell C2)  |
3   |           |           |           |
4   |           |           |           |
```

Your program will repeatedly accept commands from the user (to change values stored in cells, perform operations on cells, etc.), until the user types ‘quit’, at which point your program will end. After some commands, your program will re-print the updated spreadsheet, showing the changes caused by the user. Commands are described in detail later on in this document.

In Part B (to be assigned in Quarter 3), some commands will specify not just a single cell, but “cell ranges”. A cell range is a group of cells in a rectangular region. A1-B3, A1-A7 and A1-B1 are all valid cell ranges. Cell ranges always specify the two opposite corners of the rectangle of cells in the range, with the upper-left corner always appearing first as in the preceding examples. Details on which commands allow cell ranges are below in the Formulas section. Note that, in Part A, you will lay the groundwork for formulas, but formulas will not be evaluated until Part B.

Your spreadsheet will be able to store 3 major types of cells: empty, text, and real. All cells begin as empty cells. Text cells represent text strings that are input with double quotes surrounding them but are printed on the screen without them. Real cells contain decimal numbers or percentages, and can have formulas in them. Details on these cells are provided later on in this document.

An example (partial) spreadsheet is shown below. The following shows non-empty cells in the sheet:

- C2: the text value “hello”
- C3: the real value 2.0
- D3: the real value 4.0
- A1: the percent value 3%
- F3: the average of cell C3 and D3, represented as (avg C3-D3)
- Cell range B5-E7 has been highlighted as an example cell range.

```
   |A         |B         |C         |D         |E         |F
1  |3%        |          |          |          |          |
2  |          |          |hello     |          |          |
3  |          |          |2.0       |4.0       |          |3.0
4  |          |          |          |          |          |
5  |          |          |          |          |          |
6  |          |          |          |          |          |
7  |          |          |          |          |          |
```

Specifications

Your spreadsheet will not have a GUI (Graphical User Interface) like Microsoft Excel, but will instead be more similar to VisiCalc and will be built into the console window. To ensure that our output will fit well within the console window, our spreadsheet will only be **12 columns (A-L) by 20 rows (1-20)** (unlike the unlimited scrollable version of the modern Microsoft Excel spreadsheet program).

To facilitate design and testing aspects of the project, the starter project we provide will include classes for you to extend, or provide code for. The code you submit for each checkpoint and the final project must work with unmodified copies of those supporting files. More details appear later in this document.

Upon starting your spreadsheet program, an empty spreadsheet will be generated and displayed on the console. (See Examples section below.) After this, your program will enter a loop where it waits for a command, acts based on the command, and prints either the result of the command or the updated spreadsheet depending on the specific command. The program exits when the user enters quit.

As per the examples, each row of the spreadsheet must be numbered and each column lettered. Horizontally, cells needs to be separated by |'s and need to have equal length (**the width of each cell is 10 characters**, not including the | on each side). If the value doesn't fit, it is truncated (shortened) in the display. The internal representation of a value should be complete; values should only be truncated while being displayed in the spreadsheet. The entire value should be used whenever it is displayed on the console via a command to display only that cell, or the cell is used for a calculation, whether or not it is truncated in the spreadsheet display.

Commands

The following table lists all the possible commands that the user may enter into the program to perform some actions. Note that commands, methods, cell references, and ranges are not case-sensitive (AVG, aVg, and avg all mean the same thing, as do A5 and a5). However both case and whitespace are important in text values:

- "`Hello`" is different from "`hello`"
- "`hi fi`" is different from "`hi  fi`".

After each command is executed, either the entire spreadsheet is re-printed, or a single value is printed. The table describes which to print.

Command	Examples	Action
<cell>	`B3`	Displays the content of the cell • Empty cell: Displays the empty string • Text cell: String content *with* enclosing quotes (though when the Text cell is displayed in the sheet, quotes are not shown) • Real cell: If the cell contains a **formula**, display the formula as it was entered by the user, including the outer parentheses (not the result of the formula). **Otherwise**, display the decimal value of the cell. • The entire table should *not* be displayed afterwards

<cell> = <*value*>	`F7 = "hi"`
	`B2 = ( avg B5-D6 )`
	`E9 = 5`
	`C2 = 4.5`
	`D1 = ( 2 * 7 / 3 )`
	`D2 = ( C2 + 1 )`
	`F5 = 6.2837%`
		This sets the contents of the cell to the provided input and determines the type of the cell. In the examples shown, F7 would be a TextCell; B2, E9, C2, D1, D2, and F5 would each be a RealCell.
		• The entire table should be displayed afterwards

clear	`clear`	Clears all cells in the spreadsheet (i.e. makes them all EmptyCells) • The entire table should be displayed afterwards
clear <cell>	`clear H13`	Makes the specific cell empty • The entire table should be displayed afterwards
quit	`quit`	Quits the program. • The entire table should *not* be displayed afterwards

Text Cells

You **create** a TextCell when the assignment command specifies double-quotes around its value:

```
A1 = "zoopadawowow"
```

When the user **inspects** a TextCell, you print the complete value, with the double-quotes:

```
A1
```

causes you to print: `"zoopadawowow"`

When the **full spreadsheet prints**, the value is always truncated to fit inside the cell width.

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |zoopadawow|          |          |          |          |          |          |          |          |          |          |          |
```

Real Cells: Values, Percents, and Formulas

Real cells are the cells that hold numbers, and participate in calculations. You will create three classes to extend these: ValueCell, PercentCell, and FormulaCell.

ValueCell

You **create** a ValueCell when the assignment command specifies a simple decimal value:

```
A1 = 8.42259265958979
```

When the user **inspects** a ValueCell, you print the complete value, to full precision:

```
A1
```

causes you to print: `8.42259265958979`

When the **full spreadsheet prints**, the value is always truncated to fit inside the cell width. Notice the value is *not rounded*, it's just truncated.

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |8.42259265|          |          |          |          |          |          |          |          |          |          |          |
```

PercentCell

You **create** a PercentCell when the assignment command specifies a decimal value followed by a %

```
A1 = 8.92259265958979%
```

When the user **inspects** a PercentCell, you print the complete value, to full precision, in decimal form (not percent form):

```
A1
```

causes you to print: `0.0892259265958979`

When the **full spreadsheet prints**, the value is always printed in percent form, and the decimal portion is truncated, not rounded.

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |8%        |          |          |          |          |          |          |          |          |          |          |          |
```

FormulaCell

You **create** a FormulaCell when the assignment command specifies an expression contained in parentheses:

```
A1 = ( 1 + B5 + 3 )
```

An **arithmetic** formula is an expression involving real (Java's `double`) constants, cell references, and the operators +, -, *, and /. Order of operations must either be strictly left-to-right (no operator precedence) for full credit, or for extra credit you may follow standard operator precedence rules. Numbers in formulas can only be doubles (not fractions).

If a formula contains a cell inside it, that is called a "reference" to the cell. When a cell that is referenced by a formula is changed, the change will affect the result displayed in the formula's cell. In the above example of `A1 = ( 1 + B5 + 3 )`, if B5 has the value 0.0, then A1 would be displayed in the spreadsheet as 4.0. But if B5 is later changed to 1.5, then A1 would change to display itself as 5.5. Circular references with formulas are not permitted and will not be tested.

A formula may also be a **method** formula, as in

```
B1 = ( AVG A2-A5 )
```

A method formula can only contain one method (either `SUM` or `AVG`) and cannot contain arithmetic operations.

Some example formulas are shown below.

Input Example	Action
`B3 = ( 4 * 6 + 3 )`	Assigns formula to cell B3, which displays 27.0 when evaluated.
`A9 = ( B3 * 6 + 3 )`	Assigns formula to cell A9, which when evaluated retrieves the value of cell B3, multiplies the retrieved value by 6, and adds 3.
`L14 = ( SUM B6-C12 )`	This formula calculates the sum of the values in cell range B6-C12.
`C12 = ( AVG A1-A5 )`	This formula calculates the average of the values in cell range A1-A5.

When you parse formulas entered by the user, note that formulas always start with a left parenthesis followed by a space, all operators and operands are separated from each other with a space, and the formula ends with a space and then a right parenthesis. Although the formula is surrounded by parentheses, we will not allow parentheses inside the formula, and they will not be tested.

When the user **inspects** a FormulaCell, you print the complete formula, including the outer parentheses:

```
B1
```

causes you to print: `( AVG A2-A5 )`

In Part B, when the **full spreadsheet prints**, the formula will be evaluated, and the final value printed in the spreadsheet, truncated to fit inside the cell width. As always, the value is *not rounded*, it's just truncated.

```
   |A        |B        |C        |D        |E        |F        |G        |H        |I        |J        |K        |L        |
1  |         |13.5     |         |         |         |         |         |         |         |         |         |         |
```

Note that, in Part A, when the full spreadsheet prints, you may print any value you like for the FormulaCell, so long as it fits within the cell width, and is school appropriate.

Example

A short, basic code execution example is shown below. This is starting from the initial launch of the program. Lines entered by the user are shown below in **bold red**, larger than the program output.

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |          |          |          |          |          |          |          |          |          |          |          |          |
2  |          |          |          |          |          |          |          |          |          |          |          |          |
3  |          |          |          |          |          |          |          |          |          |          |          |          |
4  |          |          |          |          |          |          |          |          |          |          |          |          |
5  |          |          |          |          |          |          |          |          |          |          |          |          |
6  |          |          |          |          |          |          |          |          |          |          |          |          |
7  |          |          |          |          |          |          |          |          |          |          |          |          |
8  |          |          |          |          |          |          |          |          |          |          |          |          |
9  |          |          |          |          |          |          |          |          |          |          |          |          |
10 |          |          |          |          |          |          |          |          |          |          |          |          |
11 |          |          |          |          |          |          |          |          |          |          |          |          |
12 |          |          |          |          |          |          |          |          |          |          |          |          |
13 |          |          |          |          |          |          |          |          |          |          |          |          |
14 |          |          |          |          |          |          |          |          |          |          |          |          |
15 |          |          |          |          |          |          |          |          |          |          |          |          |
16 |          |          |          |          |          |          |          |          |          |          |          |          |
17 |          |          |          |          |          |          |          |          |          |          |          |          |
18 |          |          |          |          |          |          |          |          |          |          |          |          |
19 |          |          |          |          |          |          |          |          |          |          |          |          |
20 |          |          |          |          |          |          |          |          |          |          |          |          |
```

A1 = "Hello"

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |Hello     |          |          |          |          |          |          |          |          |          |          |          |
2  |          |          |          |          |          |          |          |          |          |          |          |          |
3  |          |          |          |          |          |          |          |          |          |          |          |          |
4  |          |          |          |          |          |          |          |          |          |          |          |          |
5  |          |          |          |          |          |          |          |          |          |          |          |          |
6  |          |          |          |          |          |          |          |          |          |          |          |          |
7  |          |          |          |          |          |          |          |          |          |          |          |          |
8  |          |          |          |          |          |          |          |          |          |          |          |          |
9  |          |          |          |          |          |          |          |          |          |          |          |          |
10 |          |          |          |          |          |          |          |          |          |          |          |          |
11 |          |          |          |          |          |          |          |          |          |          |          |          |
12 |          |          |          |          |          |          |          |          |          |          |          |          |
13 |          |          |          |          |          |          |          |          |          |          |          |          |
14 |          |          |          |          |          |          |          |          |          |          |          |          |
15 |          |          |          |          |          |          |          |          |          |          |          |          |
16 |          |          |          |          |          |          |          |          |          |          |          |          |
17 |          |          |          |          |          |          |          |          |          |          |          |          |
18 |          |          |          |          |          |          |          |          |          |          |          |          |
19 |          |          |          |          |          |          |          |          |          |          |          |          |
20 |          |          |          |          |          |          |          |          |          |          |          |          |
```

A1

```
"Hello"
```

B3 = 17

```
   |A         |B         |C         |D         |E         |F         |G         |H         |I         |J         |K         |L         |
1  |Hello     |          |          |          |          |          |          |          |          |          |          |          |
2  |          |          |          |          |          |          |          |          |          |          |          |          |
3  |          |17.0      |          |          |          |          |          |          |          |          |          |          |
4  |          |          |          |          |          |          |          |          |          |          |          |          |
5  |          |          |          |          |          |          |          |          |          |          |          |          |
6  |          |          |          |          |          |          |          |          |          |          |          |          |
7  |          |          |          |          |          |          |          |          |          |          |          |          |
8  |          |          |          |          |          |          |          |          |          |          |          |          |
9  |          |          |          |          |          |          |          |          |          |          |          |          |
10 |          |          |          |          |          |          |          |          |          |          |          |          |
11 |          |          |          |          |          |          |          |          |          |          |          |          |
12 |          |          |          |          |          |          |          |          |          |          |          |          |
13 |          |          |          |          |          |          |          |          |          |          |          |          |
14 |          |          |          |          |          |          |          |          |          |          |          |          |
15 |          |          |          |          |          |          |          |          |          |          |          |          |
16 |          |          |          |          |          |          |          |          |          |          |          |          |
17 |          |          |          |          |          |          |          |          |          |          |          |          |
18 |          |          |          |          |          |          |          |          |          |          |          |          |
19 |          |          |          |          |          |          |          |          |          |          |          |          |
20 |          |          |          |          |          |          |          |          |          |          |          |          |
```

A3 = "this is a really long string"

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3	this is a	17.0										
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

A3

"this is a really long string"

C3 = 12

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3	this is a	17.0	12.0									
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

C4 = 9

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3	this is a	17.0	12.0									
4			9.0									
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

B4 = 103

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Hello												
2													
3	this is a	17.0	12.0										
4		103.0	9.0										
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													

D4 = (avg B3-C4)

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Hello												
2													
3	this is a	17.0	12.0										
4		103.0	9.0	35.25									
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													

D3 = 0

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Hello												
2													
3	this is a	17.0	12.0	0.0									
4		103.0	9.0	35.25									
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													

F5 = (sum B3-D4)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3	this is a	17.0	12.0	0.0								
4		103.0	9.0	35.25								
5						176.25						
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

F5

(sum B3-D4)

D4

(avg B3-C4)

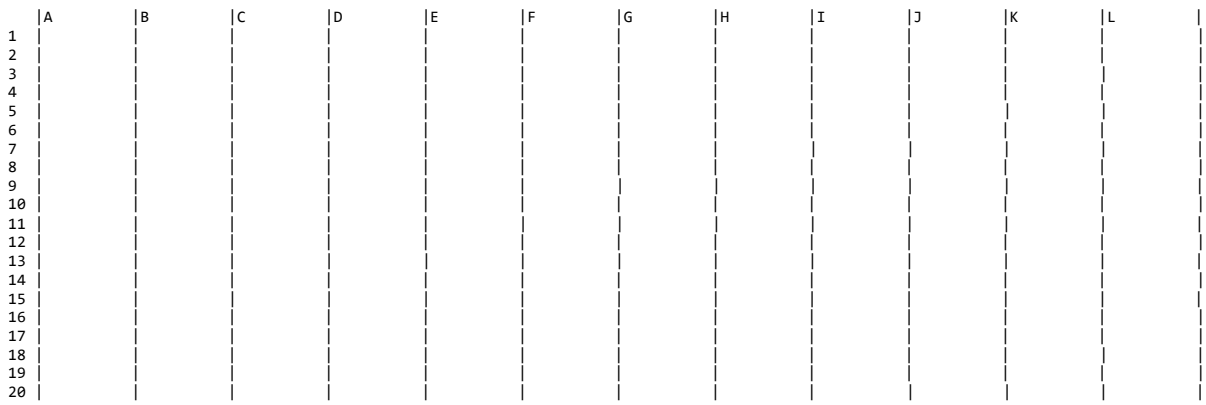
clear A3

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3		17.0	12.0	0.0								
4		103.0	9.0	35.25								
5						176.25						
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

C3 = 1000

	A	B	C	D	E	F	G	H	I	J	K	L
1	Hello											
2												
3		17.0	1000.0	0.0								
4		103.0	9.0	282.25								
5						1411.25						
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

clear



Quit