

CS 181 – Fall 2002
OUTLINE OF PROJECT #1
(discussed in detail in class)
Due Nov 1, 11:59pm

Implement simplified interpreter for SimpleSem

Implement a C++ program **simplesem** which reads SimpleSem code from a file called **code.txt** and data from a file **data.txt**. and outputs the resulting (modified) data segment to the file **data.txt** Any output to/from the **screen** is handled in a normal way.

The **format of code.txt** is:

0	statement (or empty)	For example:	0	jump 2
1	statement (or empty)		1	
2	statement (or empty)		2	set 0, D[9]

The **format of data.txt** is:

0	integer (or empty)	For example:	0	67
1	integer (or empty)		1	
2	integer (or empty)		2	-99

You are free to use **additional spaces** whenever needed.

SIMPLE VARIABLES:

Any natural number is a simple variable. If X is a simple variable then so is D[X]. Nothing else is a simple variable. For example D[D[6]] and -56 are simple variables but D[8+9] is not.

For the sake of this project, in all SimpleSem **statements all variables must be simple**. Thus, set D[D[6]], -9 is legal but set 4, 5+7 is not.

STATEMENTS:

SET:

set X, Y where X,Y are simple vars or read or write

JUMP and JUMPT

jump X where X is a simple var.

jump X, Y where X,Y are simple vars. Jump occurs only if Y evaluates to a **positive** integer value.

HALT:

halt

ADD (temporary):

add X, Y command adds Y to the existing value in memory location pointed to by X. For example,

```
set 5 , 9
set D[5], -3
add 9, 53
```

results in memory location 9 having value 50.

MULTIPLY (temporary):

multiply X, Y command multiplies the existing value in memory location pointed to by X by Y. For example,

```
set 5 , 9
set D[5], -3
multiply 9, -5
```

results in memory location 9 having value -15.

Error Checking:

The code should gracefully handle **all possible input errors** and output suitable error messages. Before any processing, the program should **check the syntax of the input files code.txt and data.txt** and output informative errors if any are found. During the run-time, the program should handle all runtime errors such as negative memory references and uninitialized memory locations. The file data.txt should be output even in the presence of runtime errors.