

Assignment 2 - Sequence Machine Compiler

ITP - 132. Written by Bryce Summers on 9.25.2020

Instructions:

- Please see the assignment instructions in the block comment at the beginning of the file smc.cpp
- Please reference this document for the errors / warnings / correct execution specification.

Updates

Updates to this document and the started code may be found and pulled from its github repository:

- https://github.com/NVCC-ITP-132-BWS/Assignment2_Sequence-Machine-Compiler

Sequence Machine Language

An ITP-132 Sequence Machine uses one string variable called the *default register* and can execute only these 5 forms of instruction:

1.		Blank line with no characters, does nothing.
2.	<code>// <Any comment></code>	This is a Comment, it does nothing.
3.	<code>read from input(); // <optional comment></code>	erases the default memory register. reads the next string token from the input stores it in the default register.
4.	<code>read from <char literal> literal; // <optional></code>	erases the default memory register. stores the character indicated by the literal in the default register as a string of length 1.
5.	<code>write to output; // <optional comment></code>	copy the value of the default register send that to the output.

Input Sequence

A Sequence Machine program may begin with an optional line that specifies its string input sequence. If this line is present, then the sequence machine program itself will begin on the second line.

The line will be the first line in the program and will contain the following tokens:

```
// IN: <string1> <string2> <string3> <string4> ... \n
```

If the input sequence line is not present, the sequence machine program begins on line 1 and is assumed to have an empty input sequence of length 0. If it is run and tries to read from the input, it will immediately raise the Input out of bounds runtime error.

Output Sequence

If there are no syntax, compile, or runtime errors, the code may be executed. Its output must be a string of the following form:

```
OUT: <out string1><out string2><out string3><out string4>...\n
```

Example Program	Input Sequence	Correct Output Sequence
<pre>read from input(); write to output; write to output;</pre>	<pre>// IN: Hello // IN: / ; // IN: print input()</pre>	<pre>OUT: HelloHello OUT: // OUT: printprint</pre>
<pre>// Split and Combine. read from input(); write to output; read from ' ' literal; write to output; read from input(); write to output; read from input(); write to output;</pre>	<pre>// IN: I a m // IN: c a t s // IN: C + + is cool! // IN: To gether apart // IN: OUT: I am // IN: IN: I a m // IN: // IN: I am // <No input line.></pre>	<pre>OUT: I am OUT: c at OUT: C ++ OUT: To getherapart OUT: OUT: Iam OUT: IN: Ia <Runtime Error, input sequence too short!> <Runtime Error, input sequence too short!> <Runtime Error, input sequence too short!></pre>

Possible Syntax Errors

Comments Rule: After the string `"/"` on a line, all later characters are ignored and cannot cause a syntax error.

Space Rule: Treat any string of multiple spaces/tabs as a one space (' '). Ex: `"read from \t \t input();"` → `"read from input();"`

id	Errors	When to Output this Error	Erroneous Example Lines
S1	First letter of the line is not recognized.	Output if the first <i>non</i> -space (' '), <i>non</i> -tab ('\t') character exists and is anything other than '/', '\\', 'r', 'R', 'w', 'W', or '\n'	<code>dead from input();</code> <code>// comment</code> <code>output << 'X';</code>
S2	Instruction not recognized. Your instruction may be missing keywords or be misspelled. (Capitalization is not the issue)	Output if neither of the following strings (in any capitalization) appear in the code part of the line: "read from ", "write to " or if the string " literal" is not 1-4 characters after the last single quote '\ '. No error if the line is blank or it only consists of tabs, /\, and/or spaces.	<code>retrieve the input();</code> <code>read from 'X';</code> <code>write data to output;</code> Not S2: <code>WriTe to OutPut;</code>
S3	Comment needs 2 slashes, not 1.	Output if a comment is written with only one '/' character.	<code>/</code> <code>/ This is a comment</code> <code>write to output; / comment</code>
S4	Comment should use '/', rather than '\'	Output if a comment is written with one or both of the '/' characters replaced by a '\' character.	<code>\ / comment</code> <code>/ \ comment</code> <code>\ \ comment</code> <code>write to output; \ \ comment</code>
S5	Add another quote to the char literal. (Literal needs terminating ' character)	Output if a first single quote character '\ ' is found and a second is not found afterwards.	<code>read from 'a literal;</code> <code>read from a' literal;</code> <code>read from '_a_ literal;</code>
S6	Backslash literal must be escaped. '\\'.	Output if "' \ '" is found. (Not in a comment)	<code>read from '\ ' literal;</code> <code>OK: // read from '\ '</code>

S7	Character literal too large. Remove character(s) from char literal.	Output if 2 single quotes are found and the string between them is length 3+ or of length 2 starting with a backslash ('\') character.	read from 'a\' literal; read from 'ab' literal; OK: read from '\n' literal;
S8	A line must end with a semicolon ';' Multi-line statements are prohibited.	Output if a line contains any uncommented non-space, non-tab characters and a semicolon is not the last non-space (' '), non-tab('\t'), non-commented character.	read from input() write to output read from 'a' literal // ;
S9	Don't capitalize syntax word(s)	Output if capital letters are found in any of the following keywords: "read", "from", "literal", "write", or "to"	Read from input(); write to output;
S10	Add a right parens ')' character	if a '(' character is found and is not followed by 0 or more spaces, then a ')' character.	read from input(;
S11	Add a left parens '(' character	if a ')' character is found and is not preceded by 0 or more spaces, then by a '(' character.	read from input);

Possible Compile Errors

Comments Rule: After the string "//" on a line, all later characters are ignored and cannot cause a compile error.

id	Errors	When to Output this Error	Erroneous Example Lines
C1	IO function variable misspelled, try writing the variable name in lowercase.	Output if the word "input" or "output" appear with any capitalized letters within.	read from Input(); read from iNPUT(); write to OUTPUT();
C2	read wants character data, not a character stream, try adding "()".	Output if neither '(', ')', follows the word "input". There can be any number of spaces / tabs in-between.	read from input; OK: read from input ()
C3	Variable name not found, try using something known like "input" or "output"	Output if the string "read from" is not followed by "input" or if the string "write to" isn't followed by "output". No error if a single quote is found.	read from coolVariable; write to cout;



Possible Runtime Errors

Comments Rule: After the string "//" on a line, all later characters are ignored and cannot cause a runtime error.

id	Errors	When to Output this Error	Erroneous Example Programs
R1	Read at least once before writing, otherwise there is nothing to write!	If the program executes the write to output command before reading data from the input or from a character literal.	write to output;
			// read from input(); write to output; read from input();
R2	Input Out of Bounds, don't read from the input when the input sequence is empty or exhausted.	Output if the program executes more read from input instructions than the length of the input sequence.	// IN: a b read from input(); write to output(); read from input(); write to output(); read from input(); write to output();

Possible Style Errors

Comments Rule: Comments **can** cause style Errors!

id	Errors	When to Output this Error	Erroneous Example Lines
L1	Do not indent Sequence Machine code.	Output if a line begins with a space (' ') character or a tab character ('\t') and the line is not blank (all spaces and/or tabs).	 read from input(); // Comment.
L2	Do not use extra spaces	If more than one space and/or tab characters are found next to each other or if a line consists of just one space and/or tab char.	read from input(); // This is my comment.  (blank line with spaces)
L3	Add one space before "//"	Output if the string "//" is not preceded by one space (' ') or the beginning of the line.	write to output;// Comment
L4	Add one space after "//"	Output if the string "//" is not followed by one space character (' ') or the end of the line.	//Comment write to output; //Comment
L5	Use space (' '), rather than tab ('\t') characters ¹ .	Output if the line contains a tab ('\t') character.	// Hello World

Erroneous Example Programs

L6	Document your code with // comments!	Output if the program contains a non-blank line of code and the string "//" is not present anywhere in the program.	read from input(); write to output();
L7	Use less blank lines, did you copy your code from a wysiwyg editor like Canvas or microsoft word into a text editor?	Output if the string "\n\n\n\n" is found in the file or if (# blank lines * 2 + 1 >= #lines and #lines > 1) (Every program has at least 1 line)	read from input(); write to output();

¹Using spaces increases the odds that your code remains nicely formatted when viewed by another programmer on another computer.

Warnings

id	Warning	When to Output this Warning	Example Programs
W1	Useless Sequence Machine Instruction, you can delete this line.	If a literal is read and another read is performed before a write. If the last instruction is a read.	read from ' ' literal; read from input(); // Overwrite. write to output;
			read from input(); write to output; read from input();
			OK: read from input(); // Skip first read from input(); // input. write to output;

id	Warning	When to Output this Error	Example Lines
W2	In many languages, the \ character means this line will continue onto the next line.	If the last character before the end of a line is a backslash (' \ '), even if it's in a comment.	read from input(); // Hello\ write to output; // World\
W3	Use Single Quotes "'" for char literals.	If the double quote character '"' is found in the code and is not sandwiched by single quotes.	read from "x" literal; OK: read from ' ' ' literal;