

Assignment 06 - Pep/9<sub>2</sub> String Sorting

(Due: Friday May 10th)

For this assignment you are given an implementation of the following Pep/9<sub>2</sub> level subprogram (in the file named `strgRead.pep2`), used within the context of the given `SSorting.pep2` source program.

```
/* Functional subprogram that returns the count of the number of non-empty
   strings which have been read and stored in the leading elements of the given
   array of strings. Note that the given limit provides the capacity of the
   array so that memory overflow does not occur. However, each string in the
   array must be a String Object that occupies 16 bytes (including the before
   byte descriptor).
*/
int strgRead(char *string[], int limit);
```

For this assignment you are to make appropriate utilization of the Pep/9<sub>2</sub> level to develop and test the following subprograms (to be stored in files named `strgPrnt.pep2` and `strgSort.pep2` respectively), also within the context of the given `SSorting.pep2` source program.

```
/* Subprogram that produces a printout of the string values stored in the leading
   elements of the given array of strings (as indicated by the value of n), one
   per line.
*/
void strgPrnt(char *string[], int n);

/* Subprogram that reorders the string values stored in the leading elements of
   the given array of strings (as indicated by the value of n), so that they are
   now in ascending lexicographic order.
*/
void strgSort(char *string[], int n);
```

This first screen shot shows sample dialogue with the `SSorting.pep2` program, as given.

Note that the Memory Dump is shown, so as to illustrate the "string values" stored in the array of strings.

Your first goal, obviously, should be to complete the `strgPrnt` subprogram so that the program may be advanced to call upon it to print the element values in the array out in the dialogue. This is shown in the next screen shot.

The screenshot shows a debugger interface with two main panes. The left pane displays the CPU registers:

	CPU		
N 0	Z 0	V 0	C 0
Accumulator	0x0005	5	
Index Register	0x0005	5	
Stack Pointer	0xFB8F	64399	
Program Counter	0x0229	553	
Instruction Specifier	00000000	stop	
Operand Specifier			
(Operand)			

The right pane shows the Memory Dump, which contains the string values entered by the user:

```

Memory Dump
0000 12 01 F5 00 18 00 05 0F ..ö.....
0008 48 65 6C 6C 6F 00 00 00 Hello...
0010 00 00 00 00 00 00 00 00 .....
0018 50 61 75 6C 00 00 00 00 Paul...
0020 00 00 00 00 00 00 00 00 .....
0028 4D 69 63 6B 61 65 6C 00 Michael...
0030 00 00 00 00 00 00 00 00 .....
0038 4A 61 63 6B 6F 77 69 74 Jackowitz...
0040 7A 00 00 00 00 00 00 0F z.....
0048 47 6F 6E 64 2D 42 79 65 Good-Bye...
0050 21 00 00 00 00 00 00 00 !.....
0058 00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 0F .....
0068 00 00 00 00 00 00 00 00 .....
0070 00 00 00 00 00 00 00 0F .....
0078 00 00 00 00 00 00 00 00 .....
0080 00 00 00 00 00 00 00 0F .....
0088 00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 00 00 00 0F .....
0098 00 00 00 00 00 00 00 00 .....
00A0 00 00 00 00 00 00 00 0F .....
00B8 00 00 00 00 00 00 00 00 .....
00B0 00 00 00 00 00 00 00 0F .....
00B8 00 00 00 00 00 00 00 00 .....
00C0 00 00 00 00 00 00 00 0F .....
00C8 00 00 00 00 00 00 00 00 .....
00D0 00 00 00 00 00 00 00 0F .....
00D8 00 00 00 00 00 00 00 00 .....
00E0 00 00 00 00 00 00 00 0F .....
00E8 00 00 00 00 00 00 00 00 .....
00F0 00 00 00 00 00 00 00 0F .....
00F8 00 00 00 00 00 00 00 00 .....

Batch I/O Terminal I/O Input/Output
PMJ's SSorting ...
Enter:Hello
Enter:Paul
Enter:Michael
Enter:Jackowitz
Enter:Good-Bye!
Enter:
5 strings read and stored
DONE!!!

```

At the bottom, there are buttons for Batch I/O, Terminal I/O, and Input/Output, and a status bar showing "Scroll to: 0x0000 SP PC".

The screenshot shows a debugger interface with two main panes. The left pane displays the CPU registers:

	CPU		
N 0	Z 1	V 0	C 1
Accumulator	0x0005	5	
Index Register	0x0005	5	
Stack Pointer	0xFB8F	64399	
Program Counter	0x024D	589	
Instruction Specifier	00000000	stop	
Operand Specifier			
(Operand)			

The right pane shows the Memory Dump, which now reflects the sorted string values:

```

Memory Dump
0000 12 01 F5 00 18 00 05 0F ..ö.....
0008 48 65 6C 6C 6F 00 00 00 Hello...
0010 00 00 00 00 00 00 00 00 .....
0018 50 61 75 6C 00 00 00 00 Paul...
0020 00 00 00 00 00 00 00 00 .....
0028 4D 69 63 6B 61 65 6C 00 Michael...
0030 00 00 00 00 00 00 00 00 .....
0038 4A 61 63 6B 6F 77 69 74 Jackowitz...
0040 7A 00 00 00 00 00 00 0F z.....
0048 47 6F 6E 64 2D 42 79 65 Good-Bye...
0050 21 00 00 00 00 00 00 00 !.....
0058 00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 0F .....
0068 00 00 00 00 00 00 00 00 .....
0070 00 00 00 00 00 00 00 0F .....
0078 00 00 00 00 00 00 00 00 .....
0080 00 00 00 00 00 00 00 0F .....
0088 00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 00 00 00 0F .....
0098 00 00 00 00 00 00 00 00 .....
00A0 00 00 00 00 00 00 00 0F .....
00B8 00 00 00 00 00 00 00 00 .....
00B0 00 00 00 00 00 00 00 0F .....
00B8 00 00 00 00 00 00 00 00 .....
00C0 00 00 00 00 00 00 00 0F .....
00C8 00 00 00 00 00 00 00 00 .....
00D0 00 00 00 00 00 00 00 0F .....
00D8 00 00 00 00 00 00 00 00 .....
00E0 00 00 00 00 00 00 00 0F .....
00E8 00 00 00 00 00 00 00 00 .....
00F0 00 00 00 00 00 00 00 0F .....
00F8 00 00 00 00 00 00 00 00 .....

Batch I/O Terminal I/O Input/Output
PMJ's SSorting ...
Enter:Hello
Enter:Paul
Enter:Michael
Enter:Jackowitz
Enter:Good-Bye!
Enter:
5 strings read and stored
0: "Hello"
1: "Paul"
2: "Michael"
3: "Jackowitz"
4: "Good-Bye!"
DONE!!!

```

At the bottom, there are buttons for Batch I/O, Terminal I/O, and Input/Output, and a status bar showing "Scroll to: 0x0000 SP PC".

Of course, after accomplishing this, you then need to focus your attention on the more challenging part of this assignment; which is to develop the `strgSort` subprogram.

The ultimate goal is to complete the program so that dialogue becomes that as shown in the final screen shot results. Note that not only are the string values now printed in ascending lexicographic order, but the Memory Dump reflects the corresponding movement of the string values among the element positions in the array of strings.

CPU				Memory Dump				
N	0	Z	1	V	0	C	1	
Accumulator	0x0005		5	0000	12 01 F5 00 18 00 05 0F		..ñ.....	
Index Register	0x0005		5	0008	47 6F 6F 64 2D 42 79 65		Good-Bye	
Stack Pointer	0xFB8F		64399	0010	21 00 00 00 00 00 00 0F		!.....	
Program Counter	0x0298		664	0018	48 65 6C 6C 6F 00 00 00		Hello...	
InstructionSpecifier	00000000	stop		0020	00 00 00 00 00 00 00 0F		.....	
OperandSpecifier				0028	4A 61 63 6B 6F 77 69 74		Jackowitz	
(Operand)				0030	7A 00 00 00 00 00 00 0F		z.....	
Batch I/O Terminal I/O				0038	4D 69 63 68 61 65 6C 00		Michael.	
Input/Output				0040	00 00 00 00 00 00 00 0F		.....	
PMJ's SSorting ...				0048	50 61 75 6C 00 00 00 00		Paul....	
Enter:Hello				0050	00 00 00 00 00 00 00 0F		.....	
Enter:Paul				0058	00 00 00 00 00 00 00 00		.....	
Enter:Michael				0060	00 00 00 00 00 00 00 0F		.....	
Enter:Jackowitz				0068	00 00 00 00 00 00 00 00		.....	
Enter:Good-Bye!				0070	00 00 00 00 00 00 00 0F		.....	
Enter:				0078	00 00 00 00 00 00 00 00		.....	
5 strings read and stored				0080	00 00 00 00 00 00 00 0F		.....	
0: "Hello"				0088	00 00 00 00 00 00 00 00		.....	
1: "Paul"				0090	00 00 00 00 00 00 00 0F		.....	
2: "Michael"				0098	00 00 00 00 00 00 00 00		.....	
3: "Jackowitz"				00A0	00 00 00 00 00 00 00 0F		.....	
4: "Good-Bye!"				00A8	00 00 00 00 00 00 00 00		.....	
--->in strgSort!				00B0	00 00 00 00 00 00 00 0F		.....	
The sorted strings are:				00B8	00 00 00 00 00 00 00 00		.....	
0: "Good-Bye!"				00C0	00 00 00 00 00 00 00 0F		.....	
1: "Hello"				00C8	00 00 00 00 00 00 00 00		.....	
2: "Jackowitz"				00D0	00 00 00 00 00 00 00 0F		.....	
3: "Michael"				00D8	00 00 00 00 00 00 00 00		.....	
4: "Paul"				00E0	00 00 00 00 00 00 00 0F		.....	
DONE!!!				00E8	00 00 00 00 00 00 00 00		.....	
				00F0	00 00 00 00 00 00 00 0F		.....	
				00F8	00 00 00 00 00 00 00 00		.....	

As your results on this assignment, you are to submit each of the following files.

<code>SSorting.pep2</code>	Revised/augmented only as needed, but with your name added in a comment at the top
<code>strgPrnt.pep2</code>	Revised as needed, and again with your name added in a comment at the top
<code>strgSort.pep2</code>	Modeled after and developed as shown in the corresponding provided source code files, and again with your name appearing in a comment at the top
<code>LastFM.docx</code>	<p>A concise report of the results of your work on this assignment, in which you explain what you have done.</p> <p>This document must contain a screen shot of the IDE that shows both; the "Source Code" pane and the "Output Pane"; similar to what is shown in this document.</p> <p>The file is required to be a MS Word readable document file, to be named in the following form, LastFM.docx (where Last is your Last Name, and F and M are your First Name and Middle Name initials; such as JackowitzPM.docx, and SmithJ.docx).</p>

Good luck,

P.M.J.