Assignment 06 – Pep/9_2 String Sorting
(Due: Friday May 10th)

For this assignment you are given an implementation of the following Pep/9_2 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_2 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.

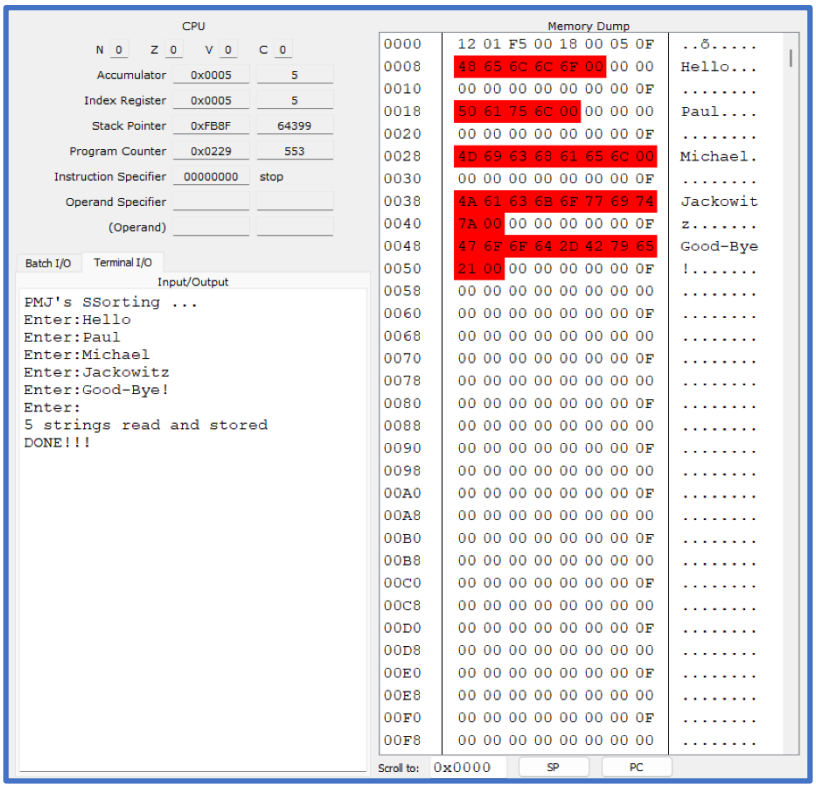
```
/* Subprogram that produces a printout of the string values stored in the leading
   elements of the given array of strings (as indicated by the value of n), one
   per line.
*/
void strgPrnt(char *string[], int n);
```

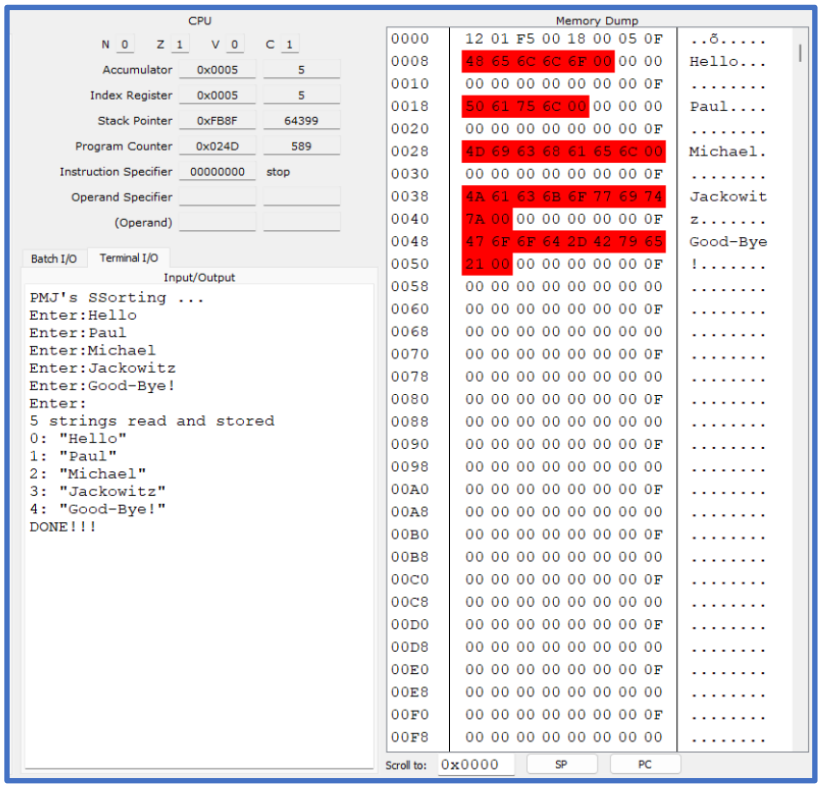
```
/* Subprogram that reorders the string values stored in the leading elements of
   the given array of strings (as indicated by the value of n), so that they are
 now in ascending lexicographic order.
*/
void strgSort(char *string[], int n);
```

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.

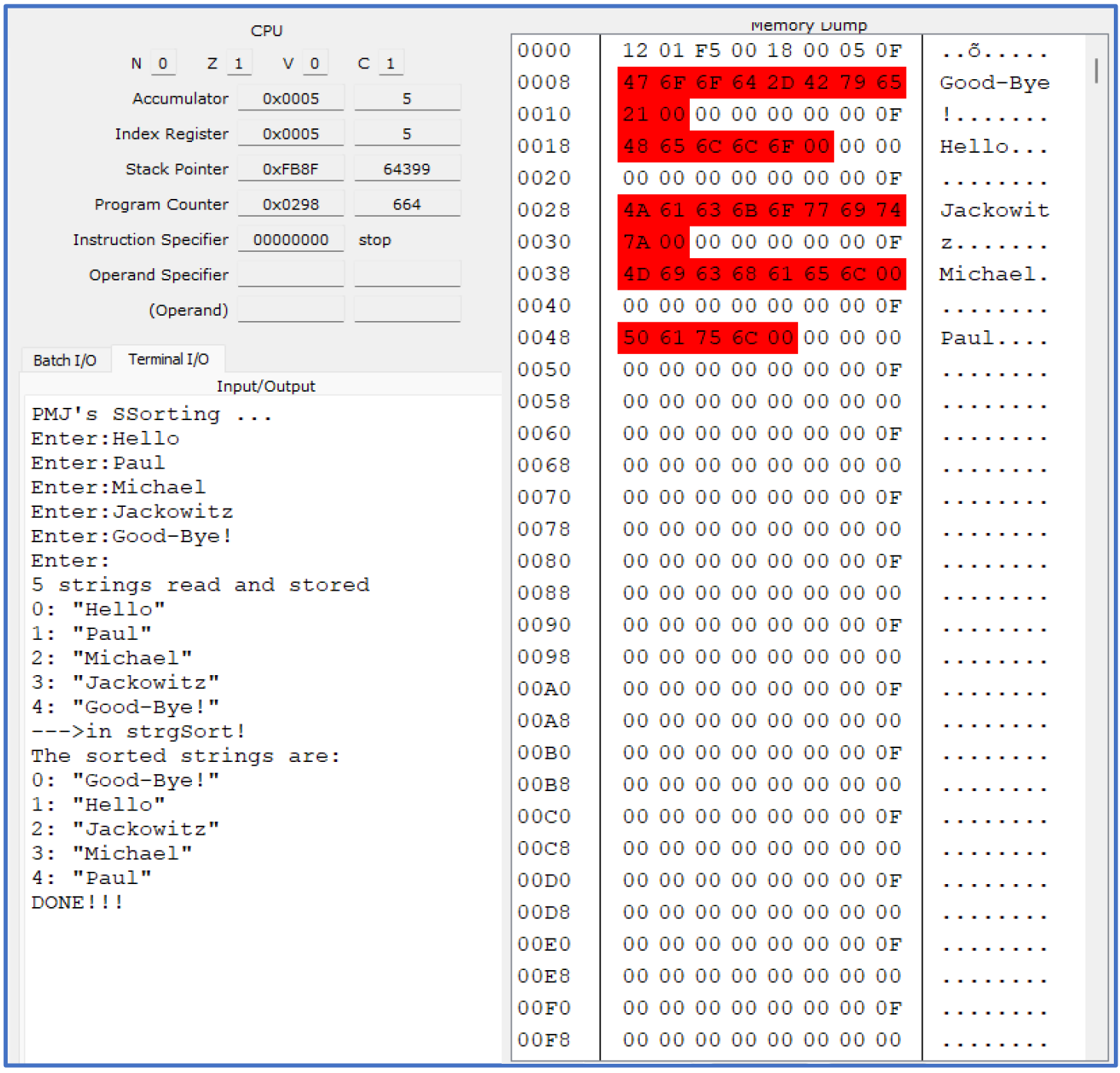
Your first goal, obviously, should be to complete the **`strgPrnt`** subprogram so that the program may be advanced to call upon it to print the element values in the array out in the dialogue. This is shown in the next screen shot.





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

N 0 Z 1 V 0 C 1		
Accumulator	0x0005	5
Index Register	0x0005	5
Stack Pointer	0xFB8F	64399
Program Counter	0x0298	664
Instruction Specifier	00000000	stop
Operand Specifier		
(Operand)		

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!"
--->in strgSort!
The sorted strings are:
0: "Good-Bye!"
1: "Hello"
2: "Jackowitz"
3: "Michael"
4: "Paul"
DONE!!!
```

Memory Dump

0000	12 01 F5 00 18 00 05 0F	..õ.....
0008	47 6F 6F 64 2D 42 79 65	Good-Bye
0010	21 00 00 00 00 00 00 0F	!.....
0018	48 65 6C 6C 6F 00 00 00	Hello...
0020	00 00 00 00 00 00 00 0F	.....
0028	4A 61 63 6B 6F 77 69 74	Jackowit
0030	7A 00 00 00 00 00 00 0F	z.....
0038	4D 69 63 68 61 65 6C 00	Michael.
0040	00 00 00 00 00 00 00 0F	.....
0048	50 61 75 6C 00 00 00 00	Paul....
0050	00 00 00 00 00 00 00 0F	.....
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	.....
00A8	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	.....

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

SSorting.pep2	Revised/augmented only as needed, but with your name added in a comment at the top
strgPrnt.pep2	Revised as needed, and again with your name added in a comment at the top
strgSort.pep2	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
***LastFM*.docx**	A concise report of the results of your work on this assignment, in which you explain what you have done. 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. 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).

Good luck,

P.M.J.