SYSC 2001 – Lab 4 – Searching a list

Part 1: Creating a flowchart

We wish to create a program that will search a list of numbers in memory (our haystack) for a particular number (our needle). If we find the needle, we should print a happy message and print the index at which we found the needle. If we search the whole list and do not find the needle, print a sad message.

Program Requirements

- Your list of 8-bit numbers should start at memory location 0000h
- Your list should be terminated with a sentinel variable (a named location in memory)
- Your messages should be '$'-terminated. All messages should start at memory location 0100h
- Your code should start at memory location 0200h
- When searching for your number, use a register as a pointer to the next number to be examined
- You cannot use an index variable (loop counter) when searching. You need to find another way to decide when you have searched all items in the list (*hint: look back at the bonus question for lab2*)
- If you find the number you're looking for, print a happy message followed by the index at which you found the needle (recall that you cannot use a loop counter in your search loop)
- If you search your whole list and fail to find the needle, print a sad message.

Deliverables

Create a flowchart illustrating your algorithm. You don't need to go into details on how you will print your messages, but you should show how you will initialize your variables, search your list, and decide when to print which message. **Stop and show a TA your flowchart before going on. Create a list of numbers and demonstrate a full pass through your flowchart in the case where you i) find your needle, and ii) fail to find your needle.**

Part 2: Coding your solution in p86 ASM

Now that you have a solid algorithm, you need to create an ASM file to implement it. The following code fragments may be of use (in randomized order):

```
FoundItMsg: .DB 'Found the needle at index $'
SUB SI,HayStack
CMP AL,'$'
DisplayPort .EQU 04E9h
HayStack:  .DB 8
           .DB 2
           .DB 4
           .DB 9
ADD AL,30H
.ORG 0100h
foundIt:              ; We get here if we found the needle
     MOV BX,FoundItMsg
     MOV DX,DisplayPort
MOV AL,[Needle]       ; Load the needle into AL for comparisons
MOV SI,HayStack       ; Create a pointer into the haystack
quit:                 ; Program done
     HLT
.END Main
```

Deliverables

Show a TA your working program. Create a HayStack list in your program. You should then be able to modify this list in memory in Virgo to demonstrate that your program works when the needle occurs in your list (at various posisitons) and when the needle is not in your list.

Hints

- If the needle appears multiple times, simply report the index of the first occurrence.
- Use the SI register to point to each number in your list
- Use the code from Lab2 as a reminder for how to print strings and numbers.