50 pts

Name: ____________________________________

Class Day / Time: ____________________________________

Due Date: ____________________________________

Assignment #8 – Assembly - Linked-List

In this assignment you will create and implement a singly linked list to store a quadword using an assembly macro, then use the linked list to store a series of positive integer numbers input from the console, search for four (4) integer numbers in the linked list and display the entire linked list. You can use the assembly code presented in the class or used in previous assignments. You will need to create the following assembly macro/procedures:

- a macro that creates a linked list to store up to 50 nodes. You will also need to create a struct to represent the node of your linked list as we describe in class.
- a procedure that searches a linked list of quadwords using a sequential search and returns in EAX the appropriate pointer for the linked list node where the key was found.
- a procedure that converts an address to an ASCII string, showing a hex decimal representation of the address. You will need to convert every hex decimal digit (4-bit) to an ASCII and store in a null terminate string for console output.
- a procedure that outputs to the console the entire linked link. You need to output the content of the node and the address store in the node (pointer to the next node).

The main program will perform the following steps:

1) Use the **macro to create** the linked list with 50 nodes.
2) Create a loop that will **read positive integer input** from the console and **add** these input numbers in the linked list. Add each new input at the end of the list. Use -1 as a sentinel value to terminate the input sequence. Ensure that the last node of the linked list point to NULL.
3) Create another loop that will allow the user to enter a **positive integer number** to be **searched** in the linked list. Output the address for the node in the linked list where the number was found. This loop will run four times.
4) Call the procedure to **output** the entire linked list.

Draw a flowchart and **write the corresponding assembly language program** in the flowchart for the program you are implementing. It is not necessary to indicate memory locations in the flowchart, but include **all the variable** declaration as part of the flow chart.

Implement the program; test the program a number of times with different input numbers. You will need to **turn in** a test run for the program as shown below.

```
Enter numbers for the linked list (use -1 to terminate input):

#1:  16
#2:  755
#3:  12
#4:  7
#5:  138
#6:  59
#7:  21
#8:  328
#9:  1572
#10:  89
#11:  -1


Enter an integer to search for:  755
755 was found at address XXXXXXXXH

Enter an integer to search for:  35
35 was not found!

Enter an integer to search for:  21
21 was found at address XXXXXXXXH

Enter an integer to search for:  2001
2001 was not found.


Content of linked list:

XXXXXXXXH:  16
XXXXXXXXH:  755
XXXXXXXXH:  12
XXXXXXXXH:  7
XXXXXXXXH:  138
XXXXXXXXH:  59
XXXXXXXXH:  21
XXXXXXXXH:  328
XXXXXXXXH:  1572
XXXXXXXXH:  89
```

Turn in (STAPLED IN THIS ORDER)

1. The **FIRST PAGE** of this assignment as a coversheet
2. Include the **flowchart** properly documented. The listing of **.asm source code** properly documented.
3. The output from the program, either pasted into .asm source code or using print screen.