ASSIGNMENT-1

[ASM'21-22]

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

Geeral instructions:

Regarding your file:

1) Submit **only running** code that you have tested before.
2) Use the provided template to write your code. Do not use any other templates.
3) Submit **only** the content of **main.asm** file. You have to press "Submit" button in the form.
4) Submit your code through this Google form link

5) **If you have trouble finding the problem with your code** (i.e. why a question won't give success), go to your project folder > Debug > testcases. The 'output' files in this folder are the expected output. Then go to your project folder > Debug > actual. The files in this folder show the output that the code you wrote has given. Compare the respective files in both folders (check spaces before and after text, uppercase and lowercase letters, symbols, etc..) to find your issue. Your actual output should be exactly that of the test case files to get Success for a question.
6) Do **not** use for ALL questions:
    a. Conditional directives, such as .if, .while, etc …
    b. MUL or SHIFT or ROTATE instructions
7) Use only instructions taken up until Lab 6.
8) Do not hard-code answers. It is clear that it will be considered unacceptable behavior.

This assignment is intended for *individual* contribution. Sharing ideas or part of answers is considered plagiarism and will not be tolerated (= ZERO mark for the entire assignment). All submissions will be checked for plagiarism automatically.

Marks will be **deducted** for not following any of the above submission rules (-5/instruction).

PLEASE WATCH THE AUTOGRADER EXPLANATION VIDEO AND DOCUMENTATION CAREFULLY BEFORE STARTING YOUR CODE

Notes:

1. All the sample runs were taken from the assignment's template.
2. Array values are not read in one line. They are read line-by-line (i.e., separated by new lines).

Good luck
Assembly team 2020

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

Q1. In the procedure called "Q1", write assembly code that swaps two characters in a word. The program should read the indices of the two characters to swap and the word. The program should display the word again with the characters swapped.

Constraints:

The string will not exceed 30 characters.

Sample Input:

Please, enter your first index: 4

Please, enter your second index: 2

Please, enter your word: Hello

Sample Output:

Your swapped word: Hlleo

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
1
Please, enter your first index: 4
Please, enter your second index: 2
Please, enter your word: Hello
Your swapped word: Hlleo
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 1 question-1 sample run

Q2. In the procedure called “Q2”, write assembly code that reads a number n from the user, then calculates and prints the Triangular Number Sequence up to the nth number. Print only one space

Good luck
Assembly team 2020

among the elements of the sequence. (DO NOT hard-code the Triangular Number Sequence in your code.)

Consider a sequence of numbers, in which the first element is 1. The second element of the sequence is the previous element plus the position of the second element (i.e., 1 + 2 = 3). The position of the first element is 1, the second element is 2, and so on. The third element of the sequence is the previous element plus the position of the third element (i.e., 3 + 3 = 6). If we continue this sequence, we get what is called the Triangular Number Sequence, as follows:

1, 3, 6, 10, 15, 21, 28, 36, 45, ...

Sample Input: 6

Sample Output: 1 3 6 10 15 21

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
2
6
1 3 6 10 15 21
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 2 question-2 sample run

Q3. In the procedure called “Q3”, write assembly code that concatenates two source arrays into a destination array. The first array is of type BYTE and the second array is of type WORD. The destination array is of type DWORD where it should hold the values of the first array and the second array concatenated.

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

Sample Input:

Elements of Array1: 6

Input Array1: 1 4 3 2 5 6

Elements of Array2: 4

Input Array2: 9 7 8 5

Sample Output:

Concatenated Array: 1 4 3 2 5 6 9 7 8 5

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
3
# Elements of Array1: 6
Input Array1: 1
4
3
2
5
6
# Elements of Array2: 4
Input Array2: 9
7
8
5
Concatenated Array: 1 4 3 2 5 6 9 7 8 5
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 3 question-3 sample run

Q4. In the procedure called "Q4", write assembly code that reads two words from the user: the first one is a correct spelling of a word, and the second one is a correct/incorrect spelling of the first word. Determine if the second word is spelled correctly. Finally, output the second word and the degree of correctness.

Good luck
Assembly team 2020

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

The degree of correctness is as follows:

“CORRECT” if it is an exact match

“ALMOST CORRECT” if no more than 2 letters are wrong

“WRONG” if 3 or more letters are wrong

Words will contain only upper-case letters. The maximum word length is 20. Each word will be on a separate line.

Sample Input:

SAMPLE

SIMPLE

THEIR

THEIR

WINDMILL

WINDOWS

ASSEMBLY

ASSEMBLYY

Sample Output:

SIMPLE IS ALMOST CORRECT

THEIR IS CORRECT

WINDOWS IS WRONG

Good luck
Assembly team 2020

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

ASSEMBLYY IS ALMOST CORRECT

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
4
SAMPLE
SIMPLE
SIMPLE IS ALMOST CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
4
THEIR
THEIR
THEIR IS CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
4
WINDMILL
WINDOWS
WINDOWS IS WRONG
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
4
ASSEMBLY
ASSEMBLYY
ASSEMBLYY IS ALMOST CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 4 question-4 sample run

Q5. In the procedure called "Q5", write assembly code that Write an assembly program that takes a binary number as a string from the user and converts it into its unsigned decimal value. Maximum length of the binary string is 32. DO NOT use ReadBin function in the Irvine library.

Hint: You can raise a number to a power using a loop of sum.

Sample Input:

Bin String: 11111111

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

Sample Output:

Decimal: 255

Sample Input:

Bin String: 11011

Sample Output:

Decimal: 27

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
5
Bin String: 11111111
Decimal: 255
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
5
Bin String: 11011
Decimal: 27
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 5 question-5 sample run

Q6. **[BONUS]** In the procedure called "Q6", write assembly code that reads a number N from the user and draw a diamond shape using that number. Its shape is defined by the following:

- The central line of the diamond contains 2N-1 asterisks.
- The difference between each row and the adjacent row is 2 asterisks.
- Spaces are only on the left of the diamond, and the difference between each row and the adjacent row is 1 space.
- Note that there is one space before the central line.

Good luck
Assembly team 2020

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

Sample Input:

4

Sample Output:

```
    *
   ***
  *****
 *******
  *****
   ***
    *
```

Sample Run:

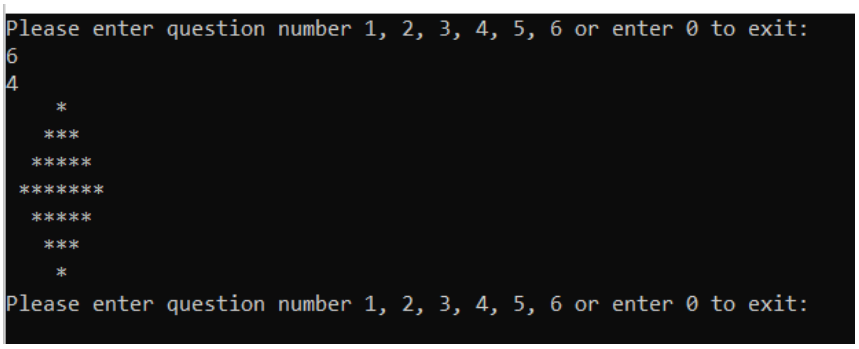

Figure 6 question-6 sample run

Hints:

- WriteInt is an Irvine function that prints an integer value that must be stored in EAX register (number's sign is printed).
- ReadInt is an Irvine function that reads an integer from the keyboard and stores it in EAX register (the input integer is signed).
- WriteDec is an Irvine function that prints an integer value that must be stored in EAX register (number's sign is not printed).
- ReadDec is an Irvine function that reads an integer from the keyboard and stores it in EAX register (the input integer is not signed).
- WriteChar is an Irvine function that prints a character that must be stored in AL register.

Good luck
Assembly team 2020

Assignment: 1 (Credit-Hour Programs)

Faculty of Computer & Information Sciences
Due date: 24 April 2022 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Salsabil Amin

- ReadString is an Irvine function that reads a string from the keyboard, stopping when the user presses the Enter key. Pass the offset of a buffer in EDX and set ECX to the maximum number of characters the user can enter. The procedure returns the count of the number of characters typed by the user in EAX.
- WriteString is an Irvine function that writes a string to the console. Pass the offset of a buffer in EDX.
- ReadHex used to read a hexadecimal value from the user. The value after the read is stored in EAX register.
- WriteHex used to write a hexadecimal value to the screen. The value to be displayed is stored in EAX register before calling this procedure.
- ReadChar is used to read a char from the console. The value read from the console is placed in "al" register.
- More about these functions and similar ones can be found in section 5.3 of the book.

Good luck
Assembly team 2020