**Assembly Programming HW2**

Assembly Programming (CSE3030)

(Spring 2020)

8th May, 2020

Instructor: Prof. Youngjae Kim

In this homework assignment, students will have to program three tasks. Please carefully read each task and program in assembly language.

**Task I**

Make a program that takes the string as input from the user and prints the reversed string as output. There are several conditions for conversion: uppercase should be changed to lowercase and lowercase should be changed to uppercase. In this assignment, students should make at least one user-defined procedure to read a string and to print the string in reverse order. An example for your program is shown in below box.

NOTE:

1. The string length does not exceed 40 characters.
2. ‘Enter’ character (<ent>) is not counted in string length.
3. If input string is larger than 40 characters, ignore the string and get input again.
4. You can freely define procedures in your convenience.
5. Input ‘Enter’ character (<ent>) to finish the program.

|  |
| --- |
| C:\>s171234↲  Type\_A\_String\_To\_Reverse: I like MUSIC.↲  Reversed\_String: .cisum EKIL i  Type\_A\_String\_To\_Reverse: I like ASSEMBLY PROGRAMMING.↲  Reversed\_String: .gnimmargorp ylbmessa EKIL i  Type\_A\_String\_To\_Reverse: I like BOTH Assembly programming and dATA STRUCTURE.↲  Type\_A\_String\_To\_Reverse:↲  Bye!  C:\> |

**Task II**

Make a program that gets a plain text and a key as input. Use the key to encrypt and decrypt the plain text by XORing each character of the key against a corresponding byte in the message.

NOTE:

1. The length of plain text does not exceed 40 characters.
2. ‘Enter’ character (<ent>) is not counted in string length.
3. The length of key does not exceed 10 characters.
4. If the length of key is less than plain text, repeat the key as many times as necessary until all plain text is translated.

Ex. Plain text : “Assembly Programming!”, Key : 5v\*qe

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | s | s | e | m | b | l | y |  | P | r | o | g | r | a | m | m | i | n | g | ! |
| 5 | v | \* | q | e | 5 | v | \* | q | e | 5 | v | \* | q | e | 5 | v | \* | q | e | 5 |

1. Print out the original text, encrypted text, and decrypted text once the user enters a text and a key.
2. Input ‘Enter’ character (<ent>) to finish the program.

|  |
| --- |
| C:\>s171234↲  Enter a plain text : This is the secret message↲  Enter a key : aG5!r3↲  Original Text : This is the secret message  Encrypted Text : (print message encrypted by given key)  Decrypted Text : This is the secret message  Enter a plain text : I am your father↲  Enter a key : 7b@p↲  Original Text : I am your father  Encrypted Text : (print message encrypted by given key)  Decrypted Text : I am your father  Enter a plain text : ↲  Bye!  C:\> |

**Task III**

Make a program that takes a string and a word as inputs from the user and searches the word in the string. If it finds the word in the string, it prints “Found”, else it prints “Not found”. Note that it is not searching for the matching characters but searching for the corresponding word. For example, suppose a string is “I am a teacher.” and a word for search is “tea”. Now the string has four words; I, am, a, teacher. Now the searched word, “tea” is not there. So, your program has to say, “Not found”.

An example for your program is shown in below box.

NOTE:

1. The string length does not exceed 40 characters.
2. ‘Enter’ character (<ent>) is not counted in string length.
3. If input string is larger than 40 characters, ignore the string and get input again.
4. You can freely define procedures in your convenience.
5. Input ‘Enter’ character (<ent>) to finish the program.

|  |
| --- |
| C:\>s171234↲  Type\_A\_String: I like MUSIC.↲  A\_Word\_for\_Search: MUSIC↲  Found  Type\_A\_String: I am a student at Sogang University.↲  A\_Word\_for\_Search: teacher↲  Not found  Type\_A\_String: I am a student at Sogang University.↲  A\_Word\_for\_Search: stud↲  Not found  Type\_A\_String: I like MUSIC.↲  A\_Word\_for\_Search: MUS↲  Not found  Type\_A\_String:↲  Bye!  C:\> |

**Submission:**

* Submission Due Date: 05/22 (Fri) 11:59 PM (Late : -10% per day, up to 3 days)
* You need to submit compressed **three code files**, each corresponds to each task.
* Name the .asm files with the last 6-digits of your student id and an underscore (‘\_’) and the number of the task. (e.g. **201234\_1.asm, 201234\_2.asm, 201234\_3asm**)
* Compress source code files into **zip format** and name it with your last 6-digits. (e.g. 201234.zip)
* Please submit your assignment under **HW2** in Assignment menu in Cyber Campus.

**Grading Policy:**

* If there is an assemble error, you will get 0 point.
* You will get minus one (-1) point for each wrong file name.
* Your program will be tested with different keys and texts.