|  |
| --- |
|  |
| **FACULTY OF ENGINEERING & COMPUTING** |

|  |  |  |
| --- | --- | --- |
| **Programme** | : | Bachelor of Software Engineering (Hons) |
| **Academic Year** | : | 2021 |
| **Module** | : | Computer Architecture |
| **Module Code** | : | CTEC 1002 |
| **Module Leader** | : | Module Leader |
| **Assignment Type** | : | Lab Assignment |
| **Intake/Group** | : | Intake/Group |
| **Distribution Date** | : | Friday, 1 April 2022 |
| **Submission Date** | : | Friday, 29 April 2022 3:00 PM |

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Student ID** | **total marks (30 %)** |
|  |  |  |
|  |  |

|  |  |
| --- | --- |
| **General Comments:** | |
|  | |
| **Assessor’s Signature**: | **Date:** |
| **Name: Saniah Sulaiman** | |

|  |
| --- |
| **General Instructions** |
| Use the following format for the preparation of the *assignment submission*.   * Paper size : A4 * Margins: left = 1.5”, right, top and bottom = 1” * Font size : 12 , Times New Roman/Arial * Line spacing : 1.5 * Text alignment : Full Justify * Number all pages sequentially * Number all Figures and Tables sequentially and refer them in the text * Binding: **staple at top left corner** of assignment submission |

|  |
| --- |
| **WARNING** |
| * + Assignments submitted after the due date will be considered late.   + Assignments submitted not later than two weeks after the due date will be marked, but the marks will be capped to a maximum of 40%.   + Assignments submitted later than two weeks will be marked, but carry zero mark.   + First City University College takes allegations of plagiarism very seriously. Submissions involving plagiarism will be marked, but given zero mark. Plagiarism is the attempt to pass off the work of another as your own. Information taken from the work of others should be acknowledged by reference to obviate the charge of copying.   + Collusion is an academic irregularity within the First City University College assessment regulations. Any student found colluding in the production of any assessment will be subject to an investigation with the imposition of any penalty deemed appropriate. Students must ensure they are familiar with the definition of collusion. |

|  |
| --- |
| Assignment Requirements |

|  |
| --- |
| Introduction |
| **Caesar Cipher**  The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his generals.  First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25. We can now represent the Caesar cipher encryption function, e(x), where x is the character we are encrypting, as:  http://practicalcryptography.com/media/latex/bdd325f4306cf573601de60e4e175dfbe7acbb14-11pt.png  Where k is the key (the shift) applied to each letter. After applying this function the result is a number which must then be translated back into a letter. The decryption function is :  http://practicalcryptography.com/media/latex/d23ecf89bd21e2afb3aadeddad53b997bddc0c5c-11pt.png  key: 2 (shift 2 times)  plaintext: wearediscoveredsaveyourself  ciphertext: ygctgfkueqxgtgfucxgaqwtugnh  Meanwhile, the decryption process will reversed the encryption process by using the same key. |
| Task Description |
| You are required to develop caesar cipher program using IA32 assembly language, do include ASCII characters of special symbols, space, letter and digit (ASCII value from 20h to 7Eh (space to ~) ). In this program, the user enters the plain text and encryption key. The encryption would be only a single ASCII character. The program able to encrypt and display the cipher text. The program also able to decrypt the encrypted text and display the plain text.  The example output as follows:  Enter the plaintext: Computer is fun!~  Enter the key: !  The encrypted message: Eqorwvgt”ku”hwp#!  The decrypted message: Computer is fun!  Encryption/Decryption process:  Assume ASCII character of space (20h), as the first character, ! (21h) is the second character, “ (22h) is the third character and etc. Thus, based on the example output given, each character in the plaintext will shifted for twice since ! symbol is used. The Table 1 given below is the list of character that can be encrypt/decrypt by the program. The program is limited to ASCII character only.    Table 1  The conversion character for the previous example output as follows:   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | C | o | m | p | u | t | e | r |  | i | s |  | f | u | n | ! | ~ | | E | q | o | r | w | v | g | t | “ | k | u | “ | h | w | p | # | ! |   The decryption process will reverse the encryption process by using the same key.  *Note*: Put your name and student ID as a comment at the top of your .asm file and follow by the program code. |
| caSubmission Requirements |
| Submit the compressed file of this documentation (fill up your name and student ID and convert to PDF format), and source code(.asm file) into the Eportal on 29th April 2022 (before 3pm). |

# Learning Outcomes

|  |  |
| --- | --- |
| **Learning Outcomes** | **Assessment** |
| Describe instruction sets and addressing modes for computer architecture |  |
| Demonstrate various programming techniques based on the instruction set and addressing modes. | Lab Assignment 1 |
| Apply suitable programming techniques in data path and control path design |  |
| Illustrate the memory and I/O organization in computer architecture |  |

**III. Assessment Marking Criteria**

This lab assignment contributes 30% to the overall marks. The assessment criteria are provided below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Area of Concern** | **Marks** | **Marking Criteria** | | | | | **Marks & Comments** |
| **Excellent** | **Good** | **Satisfactory** | **Marginal** | **Poor** |
| **Report Details** | | **80-100%** | **65-79%** | **50-64%** | **40-49%** | **0-39%** |
| **Program Functionality** | **40** | Program fulfils all the requirements and works correctly.   * plaintext from user * key from user * correct encryption method * correct decryption method | Program fulfils most of the requirements.   * Plaintext from user * Static key * Correct encryption method * correct decryption method | Program fulfils half of the requirements.   * Static plaintext * Key from user * Correct encryption method * correct decryption method | Program only fulfil a small portion of the requirement.   * Static plaintext and key * Correct encryption and decryption method | Program does not fulfil any of the requirement.   * Static plaintext and key * Error in both/either encryption and decryption method | ( /40) |
| **Output** | **10** | Produced output as required.  Output is properly formatted for better readability. | Produced output as required.  Output not formatted appropriately. | Output contains minor/some errors. | Produced inaccurate output. | Program does not execute. | ( /10) |
| **Source Code Readability** | **10** | Code is understandable, well-commented, and well-organized | Minor issues with consistent indentation, use of whitespace, naming, or general organization. Few places that could benefit from comments are missing them **or** the code is *overly* commented. | At least one major issue with indentation, whitespace, names, or organization. Complicated lines or sections of code uncommented or lacking meaningful comments. | Major problems with at least three or four of the readability subcategories. Comments are sparse | No comment provided and the code is difficult to read as no indentation or whitespaces provided at all | ( /10) |
|  |  |  |  |  |  | **Total** | **( /60) x 30/100 =** |