De La Salle University- Manila
Gokongwei College of Engineering

LBYCPA1
Programming Logic and Design Laboratory

Project Proposal

CRYPHYTONOLOGY: An Integration of Python Programming in Message Encryption and Decryption

Sean Alexander M. Morales
Princess Nicole G. Salonga

Project Description

This project is an extensive and interactive program that allows the user to simulate message encryption and decryption through various cryptographic methods included in the program. These cryptographic methods include the following: shift cipher, affine cipher, vigenère cipher, and the RSA cryptography. In each simulation, the program not only demonstrates the encryption and decryption processes but also describes the algorithm behind these processes. Also, the program also attempts to assess the understanding and knowledge of the users in basic cryptography by implementing a 'Question & Answer' component consisting of numerous questionnaires.

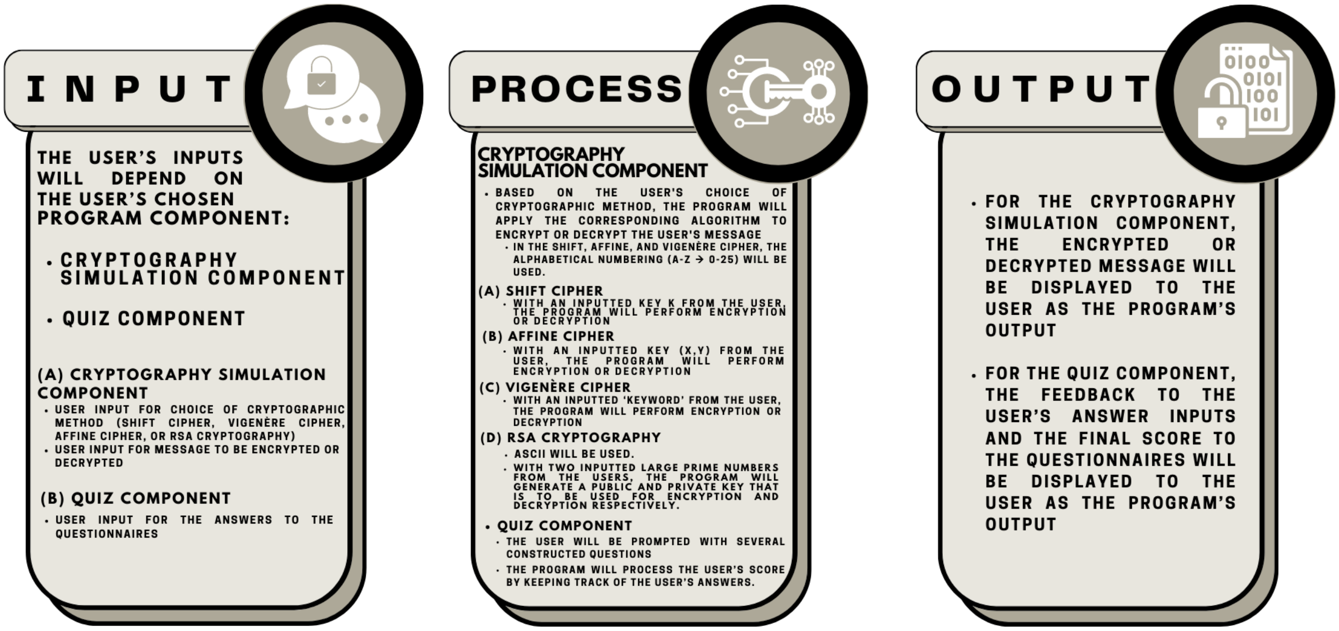
This project mainly addresses the problem of the lack of basic knowledge, understanding, and application of cryptography among individuals. In this current situation where issues on cyber security are rampant, it is a necessity to cultivate the basic knowledge and comprehension of cryptography through this proposed program.

Besides the overview and problem statement of the problem, the project aims to achieve the following technical objectives:

- To develop effective and efficient algorithms for each cryptographic method
- To ensure the functionality of the program by correctly displaying the encrypted and decrypted message
- To optimize the program's performance by implementing appropriate and efficient data structures
- To establish an operative score tracking system for the program 'Question & Answer' component
- To implement and utilize built-in Python modules in the overall program development

In developing this program, the group decided to situate each component and cryptographic methods into cases and subroutines. And within these subroutines, the algorithms and processes corresponding to the cryptographic method or component are implemented. With this process, the user can efficiently choose between options or cases that will direct to a specific component based on the user's preference.

IPO



INPUT

THE USER'S INPUTS WILL DEPEND ON THE USER'S CHOSEN PROGRAM COMPONENT:

- CRYPTOGRAPHY SIMULATION COMPONENT
- QUIZ COMPONENT

(A) CRYPTOGRAPHY SIMULATION COMPONENT

- USER INPUT FOR CHOICE OF CRYPTOGRAPHIC METHOD (SHIFT CIPHER, VIGENÈRE CIPHER, AFFINE CIPHER, OR RSA CRYPTOGRAPHY)
- USER INPUT FOR MESSAGE TO BE ENCRYPTED OR DECRYPTED

(B) QUIZ COMPONENT

- USER INPUT FOR THE ANSWERS TO THE QUESTIONNAIRES

PROCESS

CRYPTOGRAPHY SIMULATION COMPONENT

- BASED ON THE USER'S CHOICE OF CRYPTOGRAPHIC METHOD, THE PROGRAM WILL APPLY THE CORRESPONDING ALGORITHM TO ENCRYPT OR DECRYPT THE USER'S MESSAGE
  - IN THE SHIFT, AFFINE, AND VIGENÈRE CIPHER, THE ALPHABETICAL NUMBERING (A-Z → 0-25) WILL BE USED.

(A) SHIFT CIPHER

- WITH AN INPUTTED KEY K FROM THE USER, THE PROGRAM WILL PERFORM ENCRYPTION OR DECRYPTION

(B) AFFINE CIPHER

- WITH AN INPUTTED KEY (X,Y) FROM THE USER, THE PROGRAM WILL PERFORM ENCRYPTION OR DECRYPTION

(C) VIGENÈRE CIPHER

- WITH AN INPUTTED 'KEYWORD' FROM THE USER, THE PROGRAM WILL PERFORM ENCRYPTION OR DECRYPTION

(D) RSA CRYPTOGRAPHY

- ASCII WILL BE USED.
- WITH TWO INPUTTED LARGE PRIME NUMBERS FROM THE USERS, THE PROGRAM WILL GENERATE A PUBLIC AND PRIVATE KEY THAT IS TO BE USED FOR ENCRYPTION AND DECRYPTION RESPECTIVELY.

- QUIZ COMPONENT
  - THE USER WILL BE PROMPTED WITH SEVERAL CONSTRUCTED QUESTIONS
  - THE PROGRAM WILL PROCESS THE USER'S SCORE BY KEEPING TRACK OF THE USER'S ANSWERS.

OUTPUT

- FOR THE CRYPTOGRAPHY SIMULATION COMPONENT, THE ENCRYPTED OR DECRYPTED MESSAGE WILL BE DISPLAYED TO THE USER AS THE PROGRAM'S OUTPUT
- FOR THE QUIZ COMPONENT, THE FEEDBACK TO THE USER'S ANSWER INPUTS AND THE FINAL SCORE TO THE QUESTIONNAIRES WILL BE DISPLAYED TO THE USER AS THE PROGRAM'S OUTPUT

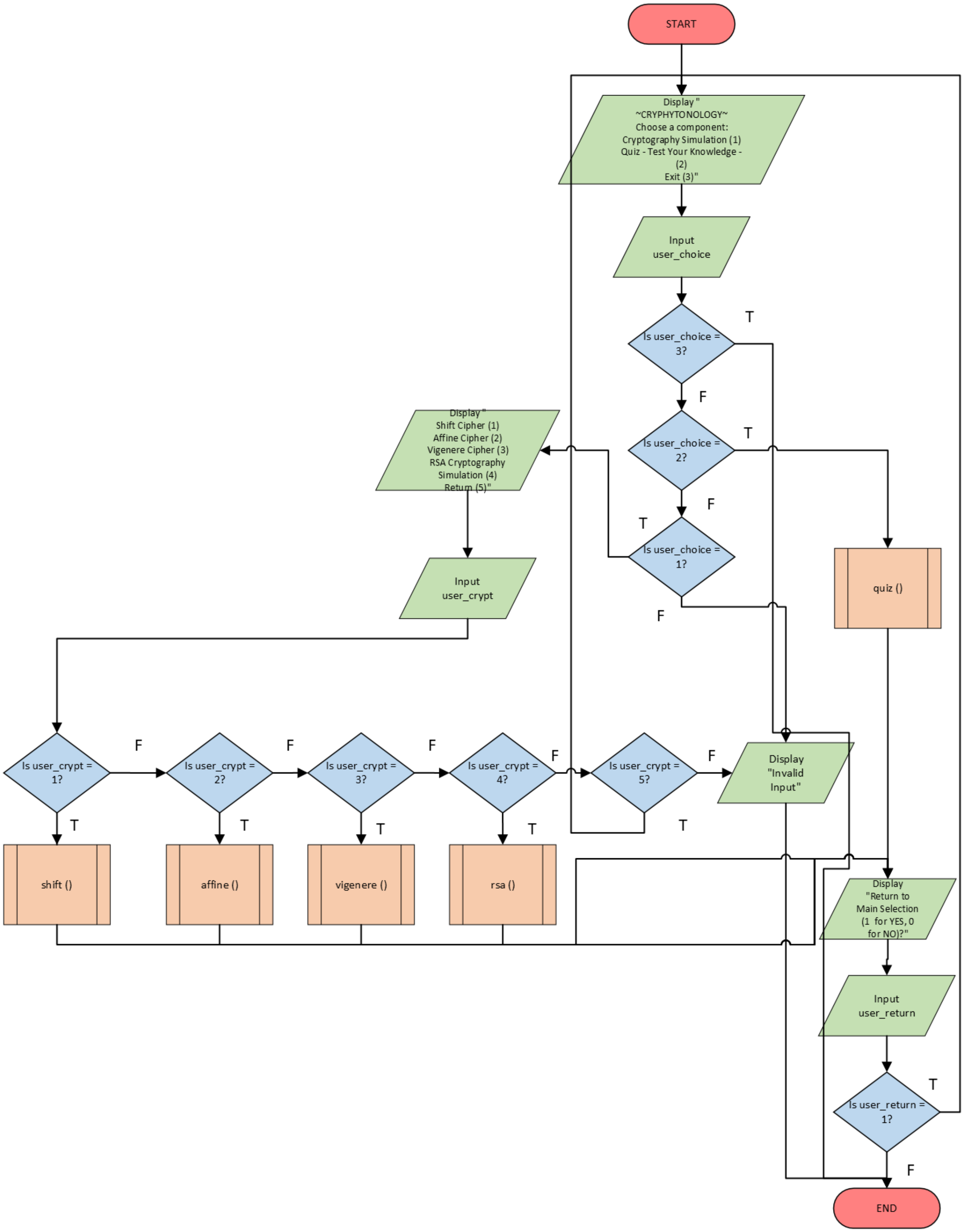
In the program's development, three key components are typically identified: Input, Process, and Output.

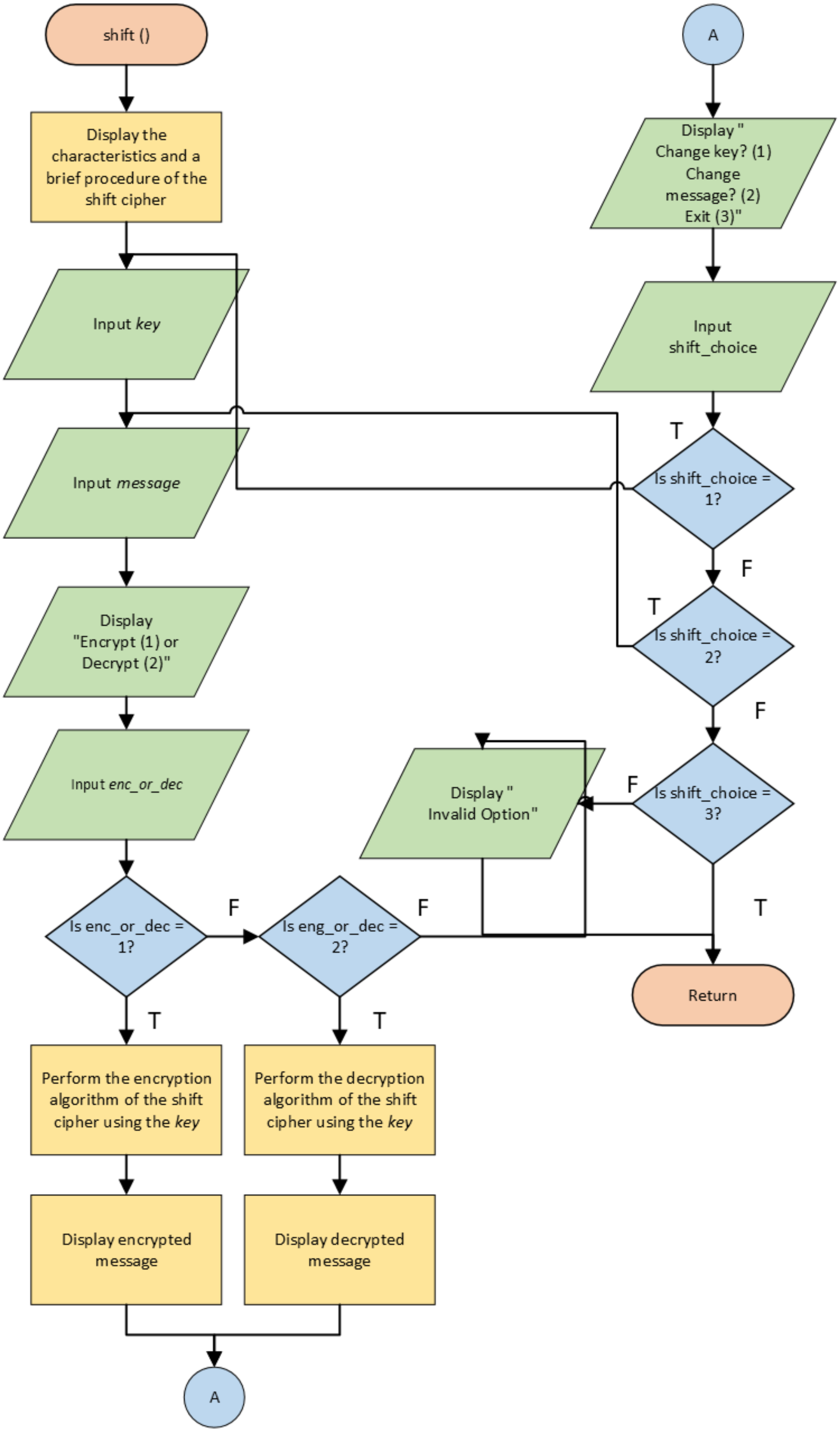
INPUT: The proposed program consists of a variety of possible inputs depending on the user's chosen component. In the 'Cryptography Simulation Component', where the simulation of cryptographic methods are set, the user's inputs include the user's preferred cryptographic method and the message that the user wants to process using that particular cryptographic method. As for the 'Quiz Component', where the questionnaires are situated, the user's inputs include the answers of the user to each question.

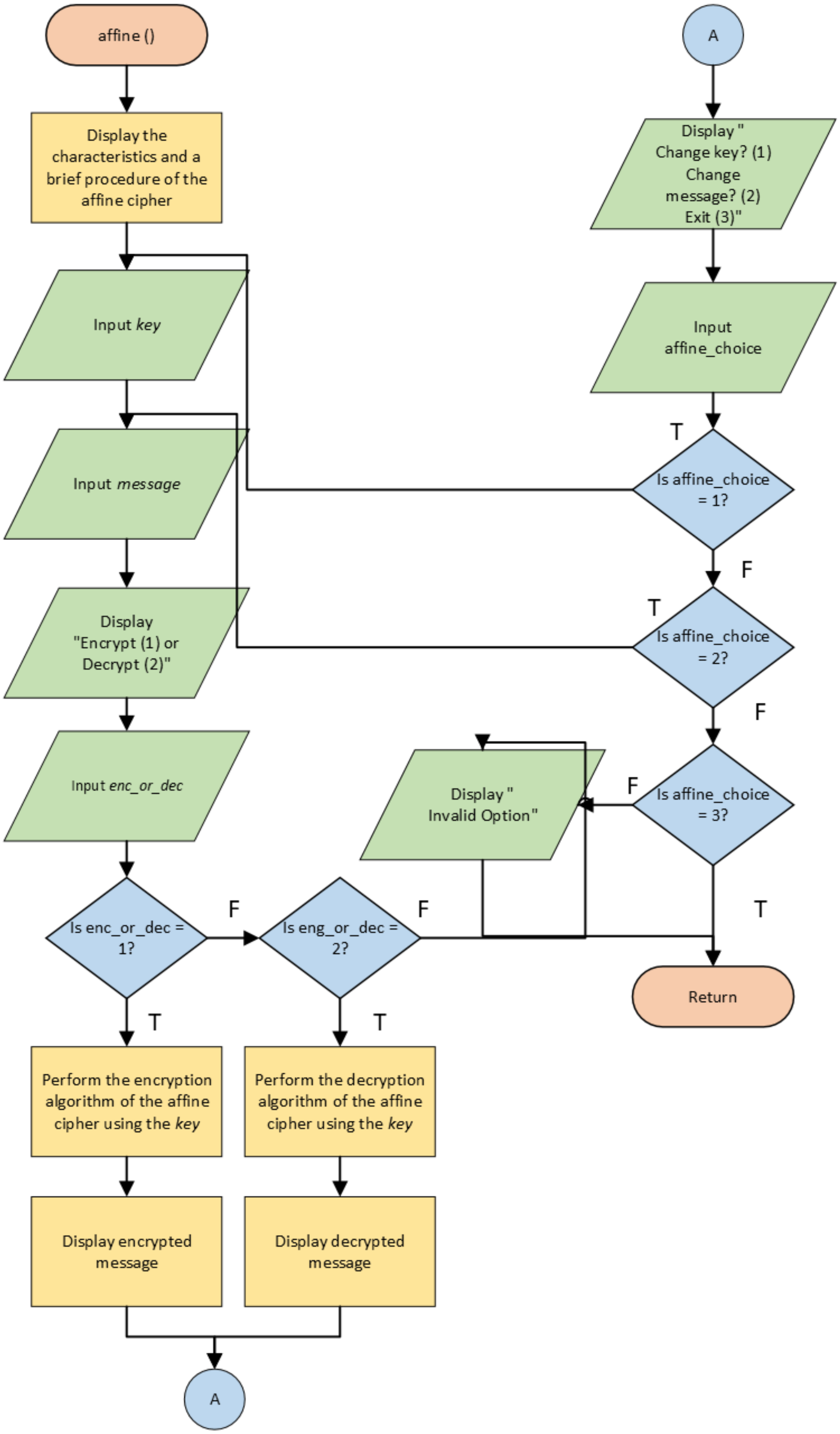
PROCESS: In each component of the program, complex processes and algorithms are performed. First, in the 'Cryptography Simulation Component', the processes involve the encryption and decryption algorithm of each cryptographic method. Furthermore, this process requires additional values and inputs from the user in order to successfully simulate the encryption and decryption processes. For the shift cipher, the algorithm requires a *key* value *K* in order to perform the cryptographic method. For the affine cipher, the algorithm also requires a *key* value to simulate the encryption and decryption process. Yet its difference with the *key* value of the shift cipher is that the affine cipher requires a tuple containing the values *x* and *y* (*x*, *y)*. For vigenère cipher, the algorithm requires a *keyword* that will serve as the basis for encryption and decryption. And for RSA cryptosystem, the algorithm requires two large prime numbers as inputs. From these inputs, the algorithm will proceed in generating the public and private keys for encryption and decryption respectively. Additionally, the mentioned cryptographic methods will incorporate the alphabetical numbering (A-Z , 0-25) and ASCII for character conversion purposes. On the other hand, the processes within the 'Quiz Component' include the continuous gathering and prompting of questions to the user. Furthermore, the embedded processes shall attentively monitor the user’s score in the questionnaire.

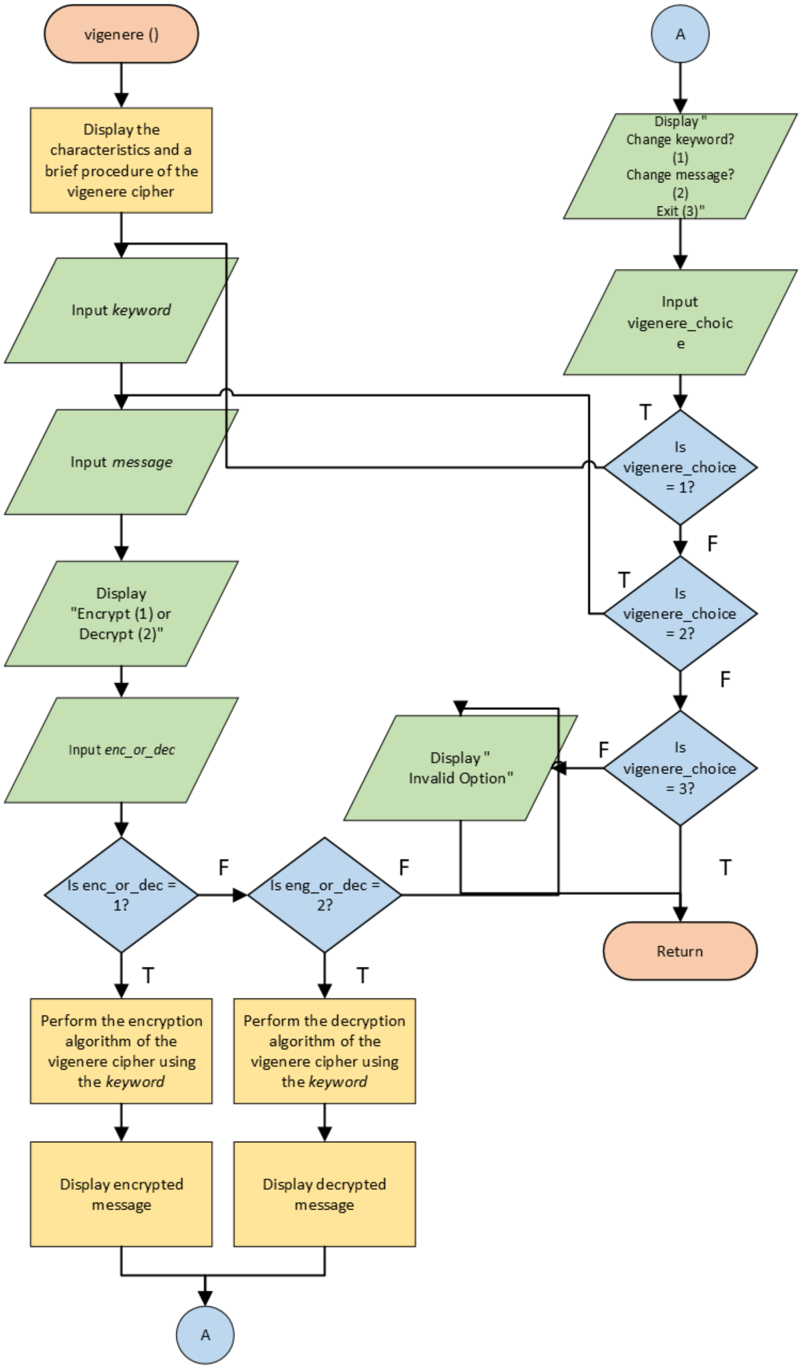
OUTPUT: Since the program has two components, two different outputs must also be generated. For the ‘Cryptography Simulation Component’, the encrypted or decrypted message serves as the program's output. On the other hand, the output for the ‘Quiz Component’ consists of the feedback to each of the user’s answers and the obtained final score from the answered questionnaire. The feedback contains the evaluation of the user’s answer and an explanation of how the correct answer was obtained.

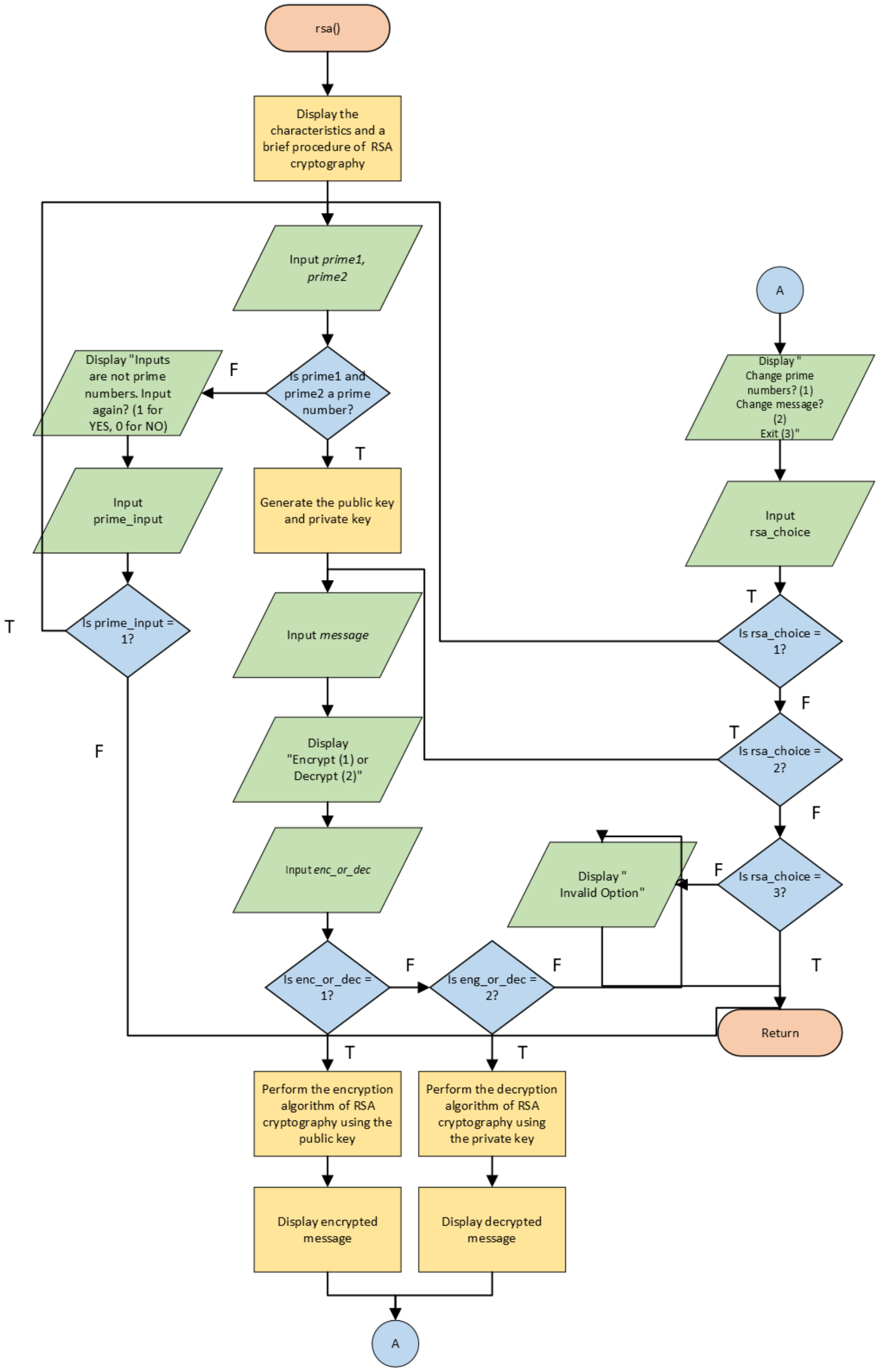
Methodology

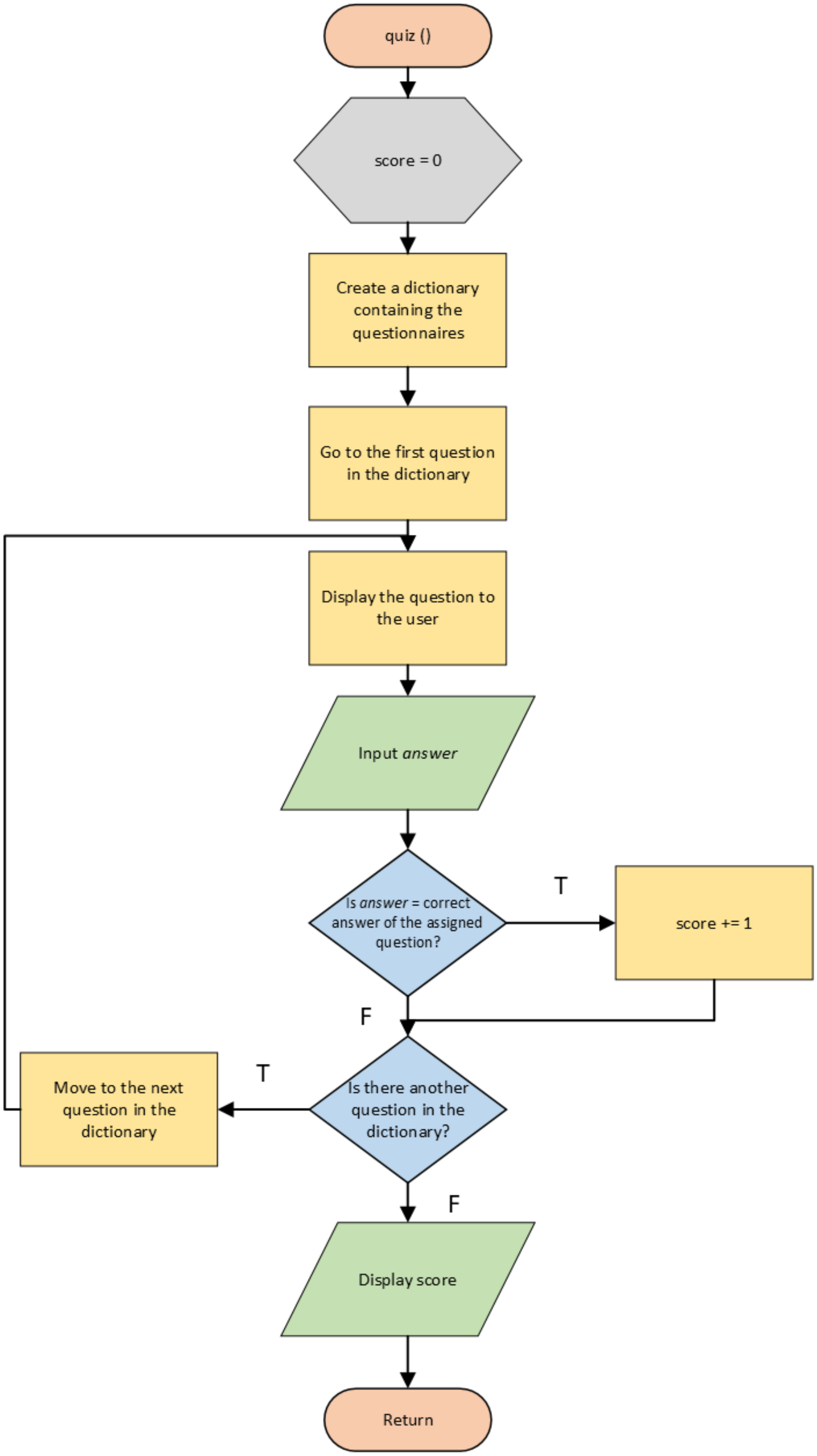



shift ()
Display the characteristics and a brief procedure of the shift cipher
Input *key*
Input *message*
Display "Encrypt (1) or Decrypt (2)"
Input *enc_or_dec*
Is enc_or_dec = 1?
F
T
Is eng_or_dec = 2?
F
T
Perform the encryption algorithm of the shift cipher using the *key*
Perform the decryption algorithm of the shift cipher using the *key*
Display encrypted message
Display decrypted message
A
A
Display " Change key? (1) Change message? (2) Exit (3)"
Input shift_choice
T
Is shift_choice = 1?
F
T
Is shift_choice = 2?
F
F
Is shift_choice = 3?
T
Display " Invalid Option"
Return


affine ()
Display the characteristics and a brief procedure of the affine cipher
Input *key*
Input *message*
Display "Encrypt (1) or Decrypt (2)"
Input *enc_or_dec*
Is enc_or_dec = 1?
F
Is eng_or_dec = 2?
F
T
Perform the encryption algorithm of the affine cipher using the *key*
T
Perform the decryption algorithm of the affine cipher using the *key*
Display encrypted message
Display decrypted message
A
A
Display " Change key? (1) Change message? (2) Exit (3)"
Input affine_choice
T
Is affine_choice = 1?
F
T
Is affine_choice = 2?
F
F
Is affine_choice = 3?
Display " Invalid Option"
T
Return

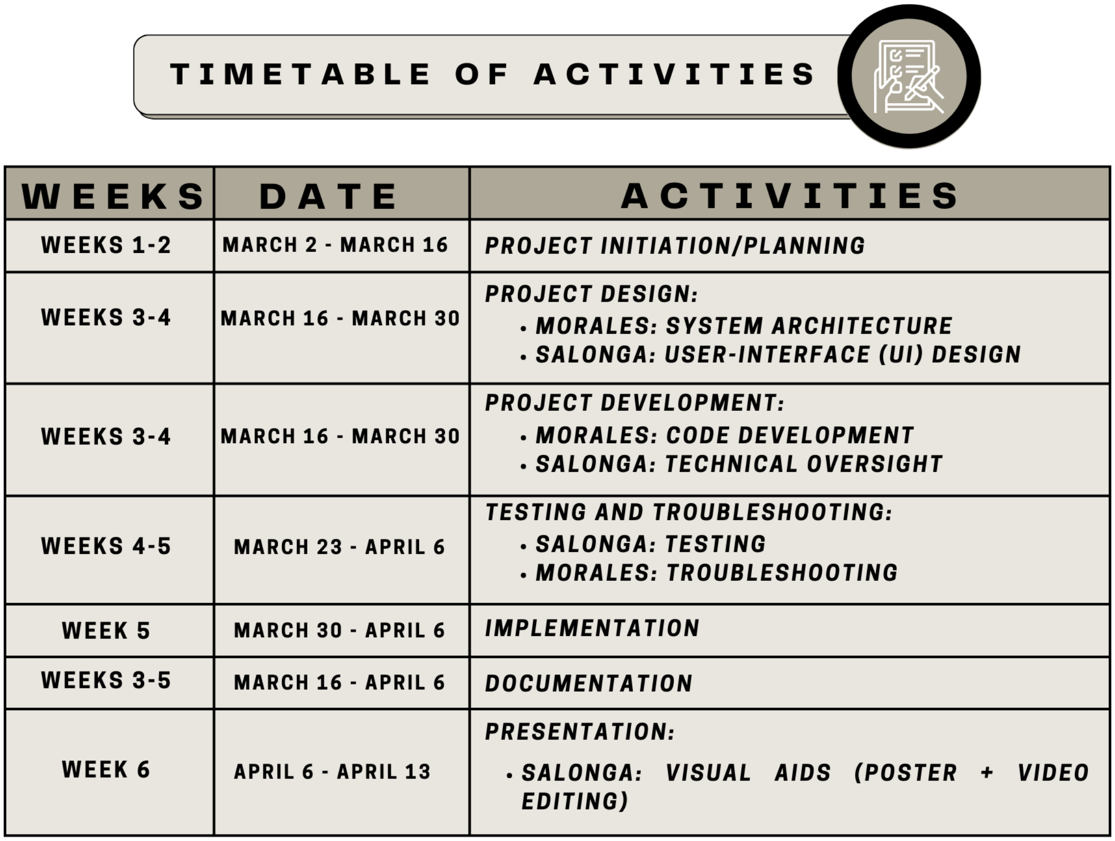

vigenere ()
Display the characteristics and a brief procedure of the vigenere cipher
Input *keyword*
Input *message*
Display "Encrypt (1) or Decrypt (2)"
Input *enc_or_dec*
Is enc_or_dec = 1?
F
Is eng_or_dec = 2?
F
T
Perform the encryption algorithm of the vigenere cipher using the *keyword*
T
Perform the decryption algorithm of the vigenere cipher using the *keyword*
Display encrypted message
Display decrypted message
A
A
Display " Change keyword? (1) Change message? (2) Exit (3)"
Input vigenere_choice
T
Is vigenere_choice = 1?
F
T
Is vigenere_choice = 2?
F
F
Is vigenere_choice = 3?
Display " Invalid Option"
T
Return


rsa()
Display the characteristics and a brief procedure of RSA cryptography
Input *prime1, prime2*
Is prime1 and prime2 a prime number?
F
Display "Inputs are not prime numbers. Input again? (1 for YES, 0 for NO)
Input prime_input
Is prime_input = 1?
T
F
T
Generate the public key and private key
Input *message*
Display "Encrypt (1) or Decrypt (2)"
Input *enc_or_dec*
Is enc_or_dec = 1?
F
Is eng_or_dec = 2?
F
T
T
Perform the encryption algorithm of RSA cryptography using the public key
Perform the decryption algorithm of RSA cryptography using the private key
Display encrypted message
Display decrypted message
A
A
Display " Change prime numbers? (1) Change message? (2) Exit (3)"
Input rsa_choice
T
Is rsa_choice = 1?
F
T
Is rsa_choice = 2?
F
Is rsa_choice = 3?
F
Display " Invalid Option"
T
Return


quiz ()
score = 0
Create a dictionary containing the questionnaires
Go to the first question in the dictionary
Display the question to the user
Input *answer*
Is *answer* = correct answer of the assigned question?
T
score += 1
F
Is there another question in the dictionary?
T
Move to the next question in the dictionary
F
Display score
Return

In the overall design and development of the project, the group seeks to apply various concepts and functions related to the Python programming language. These include Basic I/O, User-Defined Functions, Control Flow Structures, Data Structures, and Modules. Basic I/O is a necessity in the project's development as it establishes an interactive program through reading (input) and writing (output) data. As for user-defined functions, these are also essential in dividing the program into components in order to efficiently implement specific tasks in a particular component. Furthermore, these functions contribute to code reuse and easier identification. For control flow structures, these greatly provide the flow and logic of a program through the implementation of conditional and looping statements. These include if, if-elif-else, for, and while statements. In terms of data organization, the application of data structures is vital for data storage considering that the program requires multiple data. These data structures include sets, lists, dictionaries, and tuples. Lastly, the group seeks to implement built-in Python modules to perform specific processes that are beneficial for the program's development. These include the *math* module, *random* module, and *SymPy* module.

Schedule of Activities



TIMETABLE OF ACTIVITIES

WEEKS	DATE	ACTIVITIES
WEEKS 1-2	MARCH 2 - MARCH 16	*PROJECT INITIATION/PLANNING*
WEEKS 3-4	MARCH 16 - MARCH 30	*PROJECT DESIGN:* • *MORALES: SYSTEM ARCHITECTURE* • *SALONGA: USER-INTERFACE (UI) DESIGN*
WEEKS 3-4	MARCH 16 - MARCH 30	*PROJECT DEVELOPMENT:* • *MORALES: CODE DEVELOPMENT* • *SALONGA: TECHNICAL OVERSIGHT*
WEEKS 4-5	MARCH 23 - APRIL 6	*TESTING AND TROUBLESHOOTING:* • *SALONGA: TESTING* • *MORALES: TROUBLESHOOTING*
WEEK 5	MARCH 30 - APRIL 6	*IMPLEMENTATION*
WEEKS 3-5	MARCH 16 - APRIL 6	*DOCUMENTATION*
WEEK 6	APRIL 6 - APRIL 13	*PRESENTATION:* • *SALONGA: VISUAL AIDS (POSTER + VIDEO EDITING)*

References

Coghlan, N., Van Rossum, G., & Warsaw, B. (2001). *PEP 8 - Style Guide for Python Code.* Python Enhancement Proposals. https://peps.python.org/pep-0008/

Nocon, E. & Nocon, R. (2018). *Essential Mathematics for the Modern World*. C & E Publishing, Inc.

Reiter, R. (n.d.). *Intro to Python Tutorial.* Interactive Tutorials. https://www.learnpython.org/

The Asia Foundation (2022). *Cybersecurity in the Philippines: Global Context and Local Challenges.* https://asiafoundation.org/wp-content/uploads/2022/03/Cybersecurity-in-the-Philippines-Global-Context-and-Local-Challenges-.pdf

W3Schools. (2023). *Python Tutorial*. Refsnes Data. https://www.w3schools.com/python/