# Faculty of Computers and Artificial Intelligence – Cairo University

Name: Adham Hamdy Hamed Abdulhameid

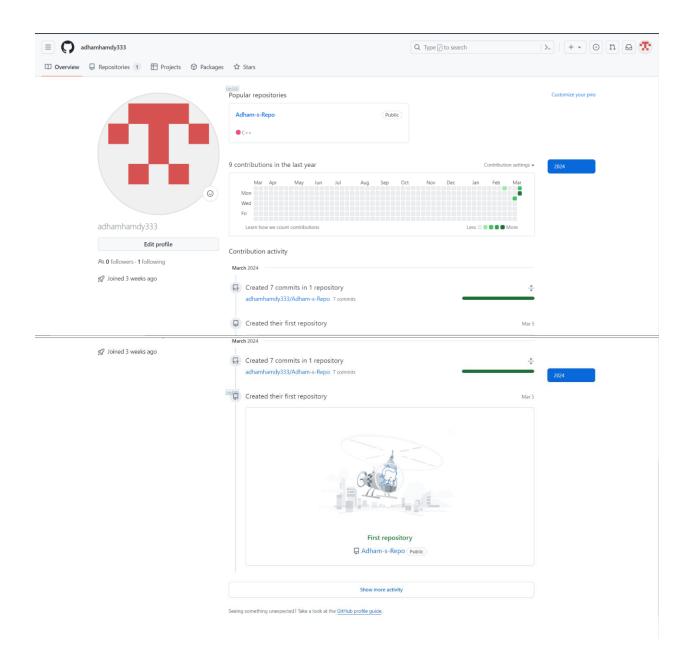
ID: 20230043

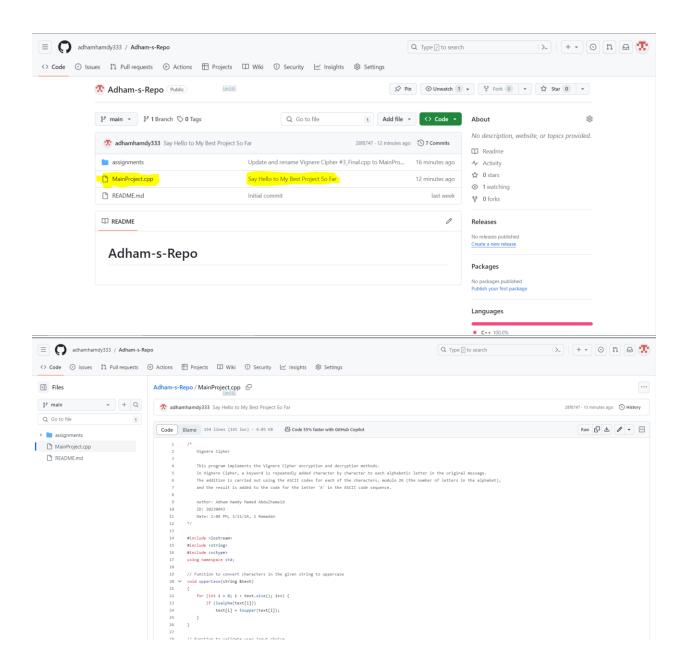
Course: CS112 - Structured Programming

Assignment #2 - Task 1

## My GitHub Profile Link:

## https://github.com/adhamhamdy333





Basic Operations that can be done on GitHub and how to do them.

## 1- Repository:

- a. Remote Repository
  - i. In the upper-right corner of any page, select (+) symbol, then click new repository.

## b. Local Repository

- Create a new folder with a specific name and assign it as your local repository and then from Cmder clone the remote repository in your local repository.
  - Go to your remote repository in github then copy its link and using Cmder with command "git clone <repo's link>" it's done.

## 2- Commit (Similar to saving a file that's been edited)

- a. Local
  - i. Firstly, add or edit the file in your local folder then, using Cmder with command "git add <file>" after adding the file now we are in Staging Area, we can commit it to local repo. using command "git commit m "message describes what happened."" it's done.

#### b. Remote

i. To make commit simply you edit the file in the remote repository then apply changes.

#### 3- Push

- a. To push your changes to the remote repository using Cmder with command "git push <RemoteName> <BranchName>"
  - i. To know what branch you're working on right now using Cmder with command "git branch" and also for remote name "git remote -v".

#### 4- Pull

a. After changes are committed on the remote repository if you already have a clone from the repository on your pc you only need to open Cmder then with command "git pull <RemoteName>" now it's done.

## 5- Pull Request

a. To make a pull request on GitHub, fork the repository, clone it to your local machine, create a new branch for your changes, and make the edits. Commit your changes, push the branch to your forked repository, and then open a pull request from your branch to the original repository, providing a descriptive message of your changes.

## Vignere Cipher Algorithm

DEFINE upperCase Function: Function to upper case the string.

- 1- Takes the string with reference.
- 2- Loops on the string if the character is alphabetic change its upper case else, ignore it.

DEFINE checkInput Function: Function to check the choice of the user from the menu.

- 1- Takes the string with reference.
- 2- While the string's size > 1 (which mean that the input its not a single character) print an exclamation message to the user and reinput the choice.
- 3- While the string is not a digit and is not one of [1, 2, 3] print an exclamation message to the user and reinput the choice.

DEFINE checkMessage Function: Function to check the inputted plain text from the user if it exceeded 80 characters or not.

- 1- Takes the string with reference.
- 2- While the string's size > 80 characters print an exclamation message to the user and reinput the plain text.

DEFINE checkKey Function: Function to check if the inputted Keyword exceeded 8 characters or not and its all alphabetical or not.

- 1- Takes the string with reference.
- 2- While the string's size > 8 characters print an exclamation message to the user and reinput the keyword.
- 3- Loops on the string to check its all alphabetical or not, if not, print an exclamation message to the user and reinput the keyword.

## DEFINE vignereCipher Function:

- 1- Takes the plain text & keyword with reference.
- 2- CALL checkMessage with plain text string as argument and checkKey with keyword as argument.
- 3- CALL upperCase for both plain text & keyword.
- 4- Assign variable keylndx to zero.
- 5- Loops on plain text if the character is alphabetical.
  - a. Assign the messageIndx variable to the addition of the ascii of the current character of plain text & and the ascii of the keyword of index (keyIndx) mod 26.
  - b. Assign the current character of plain text to the character that follows 'A'
    by messageIndx.
  - c. Assign keylndx to keylndx + 1.
  - d. Check if keyIndx = keyword's size
    - If true assign keylndx to zero. (reset keylndx)
  - e. At the end the message is finally encrypted.

### DEFINE vignereDecipher Function:

- 1- Takes the encrypted text & keyword with reference.
- 2- CALL checkMessage with decrypted text string as argument and checkKey with keyword as argument.
- 3- CALL upperCase for both encrypted text & keyword.
- 4- Assign variable keylndx to zero.
- 5- Loops on encrypted text if the character is alphabetical.
  - Assign the messageIndx variable to the subtraction between the ascii of the current character of plain text & and the ascii of the keyword of index (keyIndx) + 26 all of mod 26.
  - b. Assign the current character of encrypted text to the character that follows 'A' by messageIndx.
  - c. Assign keylndx to keylndx + 1.
  - d. Check if keyIndx = keyword's size

- i. If true assign keylndx to zero. (reset keylndx)
- e. At the end the message is finally decrypted.