# Objective

Practice the cumulative sum, char data type, for loop, and String comparison

# Problem

You want to create an app to encrypt the text messages that you send to your friends. Once your friend gets the message, the message should be decrypted so that your friend understands it. To implement this app Caesar cipher algorithm should be used. Caesar Cipher text is formed by rotating each letter by a given amount. For example, if you rotate the letter ‘A’ by 3 you should get ‘D’. rotate ‘B’ by 3 you should get ‘E’. Toward the end of the alphabet you wrap around, if for example rotate ‘X’ by 3 you should get ‘A’. rotate ‘Y’ by 3 you should get ’B’

The provided method names must be changed to meaningful names otherwise no credit will be given

# Methods

**Public static void main (String[] args)**

* Call the method1

**public static void method1()**

* Create a Scanner object
* Create a for loop to loop 20 times
  + Call the method4 to display the menu
  + Ask the user to enter the choice. the choice is either E/e to Encrypt or D/d) to Decrypt. Store it in a variable
  + If the user’s input is not a valid option, then continue with the next iteration of the loop
    - if(!(choice.equalsIgnoreCase("E") || choice.equalsIgnoreCase("D")))  
       {  
       System.out.println("Invalid Choice");  
       continue;  
       }
  + Ask the user to enter a message and store it in a variable
  + Ask the user to enter a key. The key value must be a positive number. If it is a negative number then display a message, then continue to the next iteration, similar code to the above code
  + If the user entered E or e
    - Call the method2 to encrypt the message. You need to pass the message and the key entered by the user. This method will return an encrypted message so make sure to store it in a variable
    - Display the encrypted message
  + Else if the user selected D or d
    - Call the method3 to decrypt the message entered by the user. You need to pass the message and the key to this method. This method will return a decrypted message. Make sure to store it in a variable.
    - Display the encrypted message

**Public static String method2 (String message, int key)**

**{**

1. Convert the message to uppercase using the method toUpperCase from the String class
2. Declare a variable of type string to hold the encrypted message, initialize it to “”;
3. Create a for loop to go through each letter of the message

{

Get each letter in the variable message and store it in a variable of type char use the

method call char letter = message.charAt(i) to get the letter out

if the letter is a space, replace the space with ‘#’:

{

letter = ‘#’;

}

else

{

If the letter is between ‘A’ and ‘Z’

{

Add the key to the letter: letter = letter + key

}

//checking for wrap around

If the letter is greater than ‘Z’

{

Subtract 26 from letter

}

else

{

Add 26 to letter

}

} //end of else

Add the content of the variable letter to the variable that you declared in step 2. This is a cumulative sum

}//end of the loop

Return enc

}//end of the method

**Public static String method3 (String message, int key)**

{

1. Declare a variable of type String and initialize it to “”. This variable will be used to build the decrypted message

Create a for loop to go through each letter of the message

{

Get the character at each index char letter = message.charAt(i)

If the letter is a #

letter = ‘ ‘;

If the variable letter is between ‘A’ and ‘Z’

{

Subtract the value of the variable key from the variable letter

}

If the content of the variable letter is less than ‘A’ //check for wrap around

{

Find the difference between the letter ‘A’ and the variable letter : int diff = ‘A’ – letter

letter = (char)(‘Z’ – diff + 1)

}

else if letter > ‘Z’

{

Int diff = ‘Z’ - letter

letter= (char)((‘A’ + diff + 1)

}

Concatenate the variable **letter** to the variable **that you declared in step 1.**

}//end for

Return result

}//end of the method

# Requirements

* Must provide all the methods
* Must generate the given output otherwise your code will get minimum credit
* Must follow the naming rules, and conventions
* Must follow the indentation rules
* Must change the name of all the methods to some meaningful names

# Sample output:

Refer to the provided file.