LAB1) Write a program to enter a password and then check the following:

1. Is the length of password being equal or greater than 8? If not, the program should announce that it is less than 8 and not acceptable.
2. In case of the password length ≥ 8, the program should check if it is contained small letters, capital letters, numbers, and special characters as follows:
   a. All 4 criteria: password is strong.
   b. At least hold 3 criteria: password is good.
   c. At least hold 2 criteria: password is weak.
   d. Elsewhere the password is bad

Sol.:

```
password = input('Please Enter a Password:')
capital_char = 0
small_char = 0
digit_char = 0
special_char = 0
if len(password)>=8:
    for char in password:
        if char.isupper():
            capital_char = 1
        elif char.islower():
            small_char = 1
        elif char.isdigit():
            digit_char = 1
        else:
            special_char = 1
    result = capital_char + small_char + digit_char + special_char
    if result == 4:
        print('Your Password is Strong')
    elif result == 3:
        print('Your Password is Good')
    elif result == 2:
        print('Your Password is Weak')
    else:
        print('Bad Password')
else:
    print('The Password is not Acceptable')
```

H.W.: according to the above program, do the following:

Q1) Don't use existing functions from python library (isupper, islower, etc.).

Q2) the program should warn the user if he/she use repeated letters more than 3 times.

Q3) The program should count numbers of capital, small, numbers, and special numbers.

LAB2) Write a program to do the following:

1. The program should save a list users name and their passwords.
2. The program should ask a user name and password, and then check if the user's name is entered correctly or not? If not, the program should ask the user's name continuously until it is entered correctly.
3. If the user's name is entered correctly, then the program should check if the password is entered correctly? If not, the program should give a warning to the user that he/ she has (number of remained) attempts (maximum three attempts) to enter the system.
4. In case three attempts is finished, the program should wait 60 second until the user has other three attempts to enter the correct password.

Sol.:

```
import time
user_list = ['ali', 'fatima', 'hanasa', 'halkawt']
password_list = ['abc123!', 'fat123@', 'han123#', 'hal123$']
success = 0
tries = 0
while success != 1:
    user_name = input('Please Enter Your user name:')
    user_password = input('Please Enter Your Password:')
    if user_name not in user_list:
        print('Your are not in the User list, Please Try Again!')
    elif  user_password == password_list[user_list.index(user_name)]:
        success = 1
        print('you are entered the system, welcome...')
    elif tries == 3:
        print('wrong password entered: you should wait ' +  str(60) + ' sec')
        for i in range(60, 0, -1):
            time.sleep(60)
            print(i)
        tries =0
    else:
        tries += 1
        print('you have only:'+ str(3-tries+1) + 'to enter password')
```

H.W.:

Q1) **according to the above program, do the following:** after the user is entered three wrong password and the program was entered to waiting mode (60 second). The program should give one chance to enter the password. If it was wrongly entered, then the program should enter waiting mode for 120 second, and so on. (i.e., after fifth attempt (180 second) and so on.).

Q2) Write a program to do the following:

1. The program should save a user name and it is password. And two security questions with their answers.
2. The program should ask a user name and password, and then check if the user's name is entered correctly or not? If not, the program should ask the user's name continuously until it is entered correctly.
3. If the user's name is entered correctly, then the program should check if the password is entered correctly? If not, the program should give a warning to the user that he/ she has (number of remained) attempts (maximum three attempts) to enter the system.
4. In case three attempts is finished, the program should ask the user's two security questions. If the user entered correct answer, the program should allow the user to change the password, unless the program should be ended.

LAB3) Write a program tow find greats common divisor between two number using Euclidean algorithm.

Sol.:

```
num1 = int(input('please enter first number:'))

num2 = int(input('please enter second number:'))

while num2 != 0:

    rem = num1 % num2

    num1 = num2

    num2 = rem

print('GCD=' + str(num1))
```

H.W.:

Q1) According to the above program, do the following: with finding GCD, print all equations that is founded from the division theorem between the two numbers.

Q2) use recursion function to find the GCD between two numbers.

LAB4) Write a program to find the modular multiplication inverse between two numbers using extended Euclidean algorithm.

Sol.:

```
def gcd(n1, n2):
    i = 0
    q = []
    r = []
    while n2 != 0:
        q.append(n1//n2)
        r.append(n1 % n2)
        n1 = n2
        n2 = r[i]
        i += 1
    return q, n1


num1 = int(input('please Enter the number:'))
num2 = int(input('please Enter the modular:'))
q1, gcd_r = gcd(num2, num1)
print(q1)
if gcd_r == 1:
    x_list = [0, 1]
    j = 2
    for i in range(len(q1)-1):
        x_list.append(-q1[j-2]*x_list[j-1]+x_list[j-2])
        j += 1
    print(x_list[len(x_list)-1]%num2)
else:
    print('Inverse is Not Unique')
```

H.W.: according to the above program, don't use an array or list to store all quotients and result of iterations for finding inverse number.

LAB5-I) Write a program to encrypt a message using Caesar cipher for only 26 small letters.

Sol.:

```
import string
letters = ""
cipher = ""
plain = input('please enter a plain text: ')
key = int(input('please enter the key: '))
for character in string.ascii_lowercase:
    letters = letters + character
for i in range(len(plain)):
    for j in range(len(letters)):
        if plain[i] == letters[j]:
            index = j
            c_index = (index + key) % 26
            cipher = cipher + letters[c_index]
print('cipher text= '+cipher)
```

LAB5-II) Write a program to decryption a message using Caesar cipher for only 26 small letters.

Sol.:

```
import string
letters = ""
plain = ""
cipher = input('please enter a cipher text: ')
key = int(input('please enter the key: '))
for character in string.ascii_lowercase:
    letters = letters + character
for i in range(len(cipher)):
    for j in range(len(letters)):
        if cipher[i] == letters[j]:
            index = j
            p_index = (index - key) % 26
            plain = plain + letters[p_index]
print('cipher text= '+plain)
```

H.W.:

Q1) according to above program, write a program to create two functions for encryption and decryption, and then in the main program control the function by entering en to encrypt and de to decrypt.

Q2) according to above program, use python facilities to simplify the program.

Q3) Write a program to encryption a message using Caesar cipher with ASCII code.

Q4) Write a program to decryption a message using Caesar cipher with ASCII code.