Q1) Write a program to Encrypt and Decrypt a text using Caesar cipher with only 26 English small letters?

```python
def letters_list():
    lets = []
    for i in range(ord('a'), ord('z')+1):
        lets.append(chr(i))
    return lets

def caesar_encrypt(plain_text, key):
    cipher_text = ''
    letters = letters_list()
    for plain_char in plain_text:
        index_value = letters.index(plain_char)
        cipher_text += letters[(index_value + key) % 26]
    return cipher_text


def caesar_decrypt(cipher_text, key):
    plain_text = ''
    letters = letters_list()
    for cipher_char in cipher_text:
        index_value = letters.index(cipher_char)
        plain_text += letters[(index_value - key) % 26]
    return plain_text


print(caesar_encrypt('kurdistan', 3).upper())
print(caesar_decrypt('nxuglvwdq', 3).upper())
```

Q2) Write a program to Encrypt and Decrypt a text using Caesar cipher using ASCII code?

```
def caesar_encrypt(plain_text, key):
    cipher_text = ''
    i = 0
    for plain_char in plain_text:
        index_value = ord(plain_char)
        cipher_text += chr((index_value + key) % 256)
        i += 1
    return cipher_text


def caesar_decrypt(cipher_text, key):
    plain_text = ''
    for cipher_char in cipher_text:
        index_value = ord(cipher_char)
        plain_text += chr((index_value - key) % 256)
    return plain_text


print(caesar_encrypt('kurdistan', 250).upper())
print(caesar_decrypt('EOL^CMN[H', 250).upper())
```

Homework:

1. **Write a program to Encrypt and decrypt a text using only space, numbers, capital and small letters without using ASCII code for Caesar cipher?**
2. **Control encryption and decryption (e or E: to encrypt, d or D: to decrypt) for the two pervious examples**

Q3) Write a program to encrypt and decrypt a message using Vigenere cipher using only 26 small letters?

```
def letters_list():
    lets = []
    for i in range(ord('a'), ord('z') + 1):
        lets.append(chr(i))
    return lets


def vigenere_encrypt(plain_text, key):
    cipher_text = ''
    letters = letters_list()
    key_index = []
    for key_char in key:
        for j in range(0, 26):
            if key_char == letters[j]:
               key_index.append(j)
    count = 0
    for plain_char in plain_text:
        for j in range(0, 26):
            if plain_char == letters[j]:
               cipher_text += letters[(j + key_index[count %
len(key)]) % 26]
               count+= 1
    return cipher_text


def v_decrypt(cipher_text, key):
    plain_text = ''
    letters = letters_list()
    key_index = []
    for key_char in key:
        for j in range(0, 26):
            if key_char == letters[j]:
                key_index.append(j)
    count = 0
    for cipher_char in cipher_text:
        for j in range(0, 26):
            if cipher_char == letters[j]:
                plain_text += letters[(j - key_index[count % 
len(key)]) % 26]
                count += 1
    return plain_text


print(vigenere_encrypt('thankspeshmarga', 'slemani').upper())
print(v_decrypt('lsezkfxwdlyaeos', 'slemani'))
```

HomeWorks:

1. **Write a program to encrypt and decrypt a message using Vigenere cipher using ASCII code?**
2. **Write a program to encrypt and decrypt a message using Vigenere cipher using only space, numbers, capital and small letters?**
3. **Control the previous example by entering (e or E: to encrypt, d or D: to Decrypt, any else characters: to stop the program).**

Q4) Write a program to encrypt and decrypt a message using Affine cipher using ASCII code?

```
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)


def inv(x,y):
    mod = y
    if gcd(x, y) == 1:
        a = 0
        b = 1
        while y % x != 0:
            q = y // x
            r = y % x
            ab = a - b * q
            y, x, a, b = x, r, b, ab
        return ab % mod
    else:
        return False


def affine_encrypt(plain_text, alpha, beta):
    cipher_text = ''
    for plain_char in plain_text:
        index_value = ord(plain_char)
        cipher_text += chr((index_value * alpha + beta) % 256)
    return cipher_text


def affine_decrypt(cipher_text, inv_alpha, beta):
    plain_text = ''
    for cipher_char in cipher_text:
        index_value = ord(cipher_char)
        plain_text += chr((inv_alpha*(index_value - beta)) % 256)
    return plain_text


alpha_key = int(input("Please Enter alpha: "))
beta_key = int(input("Please Enter beta: "))
if gcd(alpha_key, 256) == 1:

    flag = input("Please Enter e for encryption / d for decryption: ")
    if flag == 'e':
        plain_text = input("Please Enter a message: ")
        print(affine_encrypt(plain_text, alpha_key, beta_key))
    else:
        cipher_text = input("Please Enter a cipher message: ")
        inverse_alpha = inv(alpha_key, 256)
        print(affine_decrypt(cipher_text, inverse_alpha, beta_key))
else:
    print("Key is invalid")
```

Homework:

1. **Write a program to encrypt and decrypt a message using Affine cipher using only 26 small letters?**

Q5) Write a program to encrypt a message using Playfair cipher?

```
array = [[] for _ in range(5)]
key = input("Please Enter the key: ")
row = 0
col = 0
for i in key:
    if not any(i in array[k] for k in range(5)) and i != 'j':
        array[row].append(i)
        col += 1
    if col == 5:
        row += 1
        col = 0
letters_list = [chr(c) for c in range(97, 123)]
for i in letters_list:
    if not any(i in array[k] for k in range(5)) and i != 'j':
        array[row].append(i)
        col += 1
    if col == 5:
        row += 1
        col = 0
for row in range(len(array)):
    for col in range(len(array[row])):
        print(array[row][col], end=' ')
    print()
plain_text = input("please enter a plain text:")
cipher_text = ""
for i in range(0, len(plain_text)-1, 2):
    if plain_text[i] == plain_text[i+1]:
        plain_text = plain_text[:i+1] + 'x' + plain_text[i+1:]
if len(plain_text) % 2 == 1:
    plain_text += 'x'
print(plain_text)
index = []
for i in range(len(plain_text)):
    for row in range(len(array)):
        for col in range(len(array[row])):
            if plain_text[i] == array[row][col]:
                index.append([row, col])
print(index)
for row in range(0, len(index), 2):
    if index[row][0] == index[row + 1][0]:
        cipher_text += array[index[row][0]][(index[row][1]+1) % 5] + array[index[row+1][0]][(index[row+1][1]+1) % 5]
    elif index[row][1] == index[row + 1][1]:
        cipher_text += array[(index[row][0]+1) % 5][index[row][1]] + array[(index[row+1][0]+1) % 5][index[row+1][1]]
    else:
        cipher_text += array[index[row][0]][index[row+1][1]] + array[index[row+1][0]][index[row][1]]
print('cipher text= ' + cipher_text)
```

HomeWorks:

1. In Playfair cipher the letter j should be collect with the letter (i), the previous program did not take the letter (j) is his consideration, you should try to fix this issue?
2. Write a program to Decrypt a cipher message using Playfair cipher?