Group members:

Roll no:2	Name:Mustafa Nasikwala	Srn:202201984
Roll no:4	Name:Subodh Ugile	Srn:202201987
Roll no:14	Name: Shivam Chilkewar	Srn:202202051
Roll no:23	Name: Harshwardhan Zurunge	Srn:202202094

Code explanation:

1. Importing Libraries

```
python
import tkinter as tk
import math
```

- tkinter is Python's standard GUI (Graphical User Interface) library.
- math provides mathematical functions.

2. Variable Declarations

```
p, q, n, t, flag, e, d, temp, j, m, en, i = 0, 0, 0, 0, 0, [0] * 100, [0] * 100, [0] * 100, 0, [0] * 100, [0] * 100, 0
```

These variables are used for RSA encryption and decryption. They are initialized with default values.

3. Prime Number Check Function

```
def prime(pr):
    j = int(math.sqrt(pr))
```

```
for i in range(2, j+1):
    if pr % i == 0:
        return 0
return 1
```

- This function checks if a number is prime. It takes pr as an argument.
- It iterates from 2 up to the square root of pr and checks if pr is divisible by any number in that range.
- If it finds any divisor, it returns 0 (indicating not a prime), otherwise it returns 1 (indicating prime).

4. Calculate Possible Values of e and d

```
def ce():
    global j
    k = 0
    for i in range(2, t):
        if t % i == 0:
            continue
        flag = prime(i)
        if flag == 1 and i != p and i != q:
            e[k] = i
            flag = cd(e[k])
            if flag > 0:
                d[k] = flag
                k += 1
            if k == 99:
                break
```

- This function calculates possible values of e and d for encryption and decryption.
- It iterates from 2 to t (the totient function).
- If a number is coprime with t, it checks if it is not equal to p and q, and then assigns it to e.

- It then calculates the corresponding d using the cd function.

5. Calculate d using Extended Euclidean Algorithm

```
def cd(x):
    k=1
    while True:
        k=k+t
        if k%x==0:
            return int(k/x)
```

- This function uses the Extended Euclidean Algorithm to find the modular multiplicative inverse of x modulo t. It calculates d for encryption and decryption.

6. Encryption Function

```
def encrypt(cardNumber, len):
    global i
    key=e[0]
    i=0

    while i!=len:
        pt=cardNumber%10
        cardNumber//=10
        k=1
        for j in range(key):
            k=k*pt
            k=k%n
        temp[i]=k
        ct=k+48
```

```
    en[i]=ct
    i+=1


en[i]=-1
```

- This function performs encryption.

- It takes the cardNumber and its length (len) as arguments.

- It iterates through each digit of the card number, calculates the power using the public key e, and stores the result in en.

7. Decryption Function

```
def decrypt(len):
    global i
    key=d[0]
    i=0


while en[i] != -1:
    ct=temp[i]
    k=1
    for j in range(key):
        k=k*ct
        k=k%n
    pt=k
    m[i]=pt
    i+=1


m[i]=-1
```

- This function performs decryption.

- It takes the length of the data (len) as an argument.

- It iterates through each encrypted digit, calculates the power using the private key d, and stores the result in m.

8. Main Function

```
def main():
    global p, q, n, t, i


     (User Input) Prime Numbers p and q
    p = int(input("\nENTER FIRST PRIME NUMBER\n"))
    flag = prime(p)


    if flag == 0:
        print("\nWRONG INPUT")
        exit(1)


    q = int(input("\nENTER ANOTHER PRIME NUMBER\n"))
    flag = prime(q)


    if flag == 0 or p == q:
        print("\nWRONG INPUT")
        exit(1)


    cardNumber = int(input("\nEnter your 6-digit credit card number: "))
    expiration = int(input("\nEnter expiration month and year (YY): "))
    cvv = int(input("\nEnter CVV: "))


     Initialize arrays
    for i in range(100):
        m[i] = 0
```

```
for i in range(100):
    en[i] = 0


 Calculate n and t
n = p * q
t = (p - 1) * (q - 1)


 Calculate possible values of e and d
ce()


print("\nPOSSIBLE VALUES OF e AND d ARE")
for i in range(j - 1):
    print("\n", e[i], "\t", d[i])


 Encrypt and Decrypt
encrypt(cardNumber, 6)
decrypt(6)


encrypt(expiration, 2)
decrypt(2)


encrypt(cvv, 3)
decrypt(3)



if __name__ == "__main__":
    main()
```

- The main function handles the overall flow of the program.
- It prompts the user for prime numbers p and q, as well as the credit card details.

- It then calculates n and t and calls the functions to calculate possible values of e and d.
- After that, it performs encryption and decryption for the given credit card information.