Screenshot Code:

C: > Users > irine > OneDrive > Desktop > python2.py > main

```python
def encrypt(key, plaintext, matrix=None, ciphertext=''):
    while len(message) > 0:
        chunk_size = min(max_col_length, len(message))
        current_chunk = message[:chunk_size]
        message = message[chunk_size:]
        row = list(current_chunk) + ['_'] * (max_col_length - len(current_chunk))
        matrix.append(row)

    matrix = [[row[i] for row in matrix if row[i] != "_"] for i in range(len(matrix[0]))]

    for col in matrix:
        for char in col:
            ciphertext += char

    return ciphertext.replace(" ", "_")

def decrypt(key, ciphertext, matrix=None, position=0, plaintext=''):
    if matrix is None:
        matrix = []

    num_rows = len(ciphertext) // key
    extra_chars = len(ciphertext) % key

    for i in range(key):
        col_length = num_rows + 1 if i < extra_chars else num_rows
        current_col = list(ciphertext[position:position + col_length])
        position += col_length
        matrix.append(current_col)

    matrix = [[row[i] for row in matrix if i < len(row) and row[i] != "_"]
              for i in range(num_rows + 1)]

    for row in matrix:
        for char in row:
            plaintext += char

    return plaintext.replace("_", " ")

def main():
    plaintext = input("Enter the plaintext message: ")
    key = int(input("Enter a key: "))

    encrypted_text = encrypt(key, plaintext)
    print("Encrypted Text:", encrypted_text)

    decrypted_text = decrypt(key, encrypted_text)
    print("Decrypted Text:", decrypted_text)

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

Written Code:

```
def encrypt(key, plaintext, matrix=None, ciphertext=''):
    if matrix is None:
        matrix = []

    max_col_length = len(plaintext[:key])

    message = plaintext
    while len(message) > 0:
        chunk_size = min(max_col_length, len(message))
        current_chunk = message[:chunk_size]
        message = message[chunk_size:]
        row = list(current_chunk) + ['_'] * (max_col_length - len(current_chunk))
        matrix.append(row)

    matrix = [[row[i] for row in matrix if row[i] != "_"] for i in range(len(matrix[0]))]

    for col in matrix:
        for char in col:
            ciphertext += char

    return ciphertext.replace(" ", "_")

def decrypt(key, ciphertext, matrix=None, position=0, plaintext=''):
    if matrix is None:
        matrix = []

    num_rows = len(ciphertext) // key
    extra_chars = len(ciphertext) % key

    for i in range(key):
        col_length = num_rows + 1 if i < extra_chars else num_rows
        current_col = list(ciphertext[position:position + col_length])
        position += col_length
        matrix.append(current_col)

    matrix = [[row[i] for row in matrix if i < len(row) and row[i] != "_"]
              for i in range(num_rows + 1)]

    for row in matrix:
        for char in row:
            plaintext += char

    return plaintext.replace("_", " ")

def main():
    plaintext = input("Enter the plaintext message: ")
    key = int(input("Enter a key: "))

    encrypted_text = encrypt(key, plaintext)
```

```
    print("Encrypted Text:", encrypted_text)

    decrypted_text = decrypt(key, encrypted_text)
    print("Decrypted Text:", decrypted_text)

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

Output: