

CODE COMPLETION

```
Enter plaintext: angeline minors britney islo
Enter key: 4
```

```
Encryption Table:
```

```
A N G E
L I N E
M I N O
R A S B
R I T N
E Y I S
L O - -
```

```
Encrypted text: ALMRRELNIIAIYOGNNSTIEEOBNS
```

```
Do you want to decrypt this message? (Y/N):
```

```
Do you want to decrypt this message? (Y/N): y
```

```
Decryption Table:
```

```
A N G E
L I N E
M I N O
R A S B
R I T N
E Y I S
L O - -
```

```
Decrypted text: ANGELINEMINORASBRITNEYISLO
```

```
Do you want to encrypt another message? (Y/N):
```

```
Do you want to encrypt another message? (Y/N): y
```

```
Enter plaintext: hello123
```

```
Invalid input, alphabets only.
```

```
Enter plaintext:
```

If the user input numbers, the code will trap it.

```
Do you want to encrypt another message? (Y/N): n
```

```
Goodbye!
```

```
PS C:\Users\BISU\Documents\islo_minoras>
```

End of the program.

This Java program implements a tabular transposition cipher that enables users to encrypt and decrypt messages using a key-based matrix arrangement. The core idea is to organize the plaintext into rows and columns, then read the characters column-wise to create the ciphertext. The program begins by converting the plaintext to uppercase and removing any spaces. It then fills a two-dimensional character array with the plaintext characters, padding empty cells with a placeholder character '-'. For visualization, these placeholders are displayed as random letters when printing the encryption table, but they are not included in the final ciphertext.

The encryption process involves reading the characters from the table column by column to generate the encrypted string. Decryption reverses this process: it calculates the appropriate number of rows and columns based on the ciphertext length and the key, splits the ciphertext accordingly, and reconstructs the original plaintext by reading the characters row-wise. The program provides visual feedback by printing both the encryption and decryption tables to help users understand the transformation.

Input validation is an important feature of this program. It ensures that the plaintext consists only of alphabetic characters, rejecting any input containing numbers, spaces, or special characters. The key must be a positive integer, and invalid keys trigger error messages prompting the user to try again.

The user interacts with the program through a console interface. Upon running, the user is prompted to enter plaintext and a key. After encryption, the ciphertext is displayed, and the user is given the option to decrypt the message immediately. The program supports repeated encryption sessions until the user decides to exit. Throughout the execution, the program provides clear instructions and feedback to guide the user.

Overall, this program provides a straightforward implementation of a classical transposition cipher, illustrating fundamental concepts of text manipulation, array handling, and user input validation in Java.