S - AES

Description.

This software application encrypts messages using a simplified version of the AES method.

Files for demonstration.

In order to demonstrate the applicability of the program, it is necessary to create a text file containing the message or plain text to be encrypted.

User's guide.

In order to execute the program it is necessary to compile and run it through command line. For executing it, it is necessary to input three arguments through command line. The first one is a character that indicates what type of ***key*** is being entered (string of two characters or a 16-bit binary number) and the second one represents the ***key*** that will be used for the encryption. The first argument can be "s" (for the string of two characters) or "b"(for the binary number). The third argument represents the name of the text file that wants to be encrypted or decrypted

Command arguments example:

`"s ab message4.txt"` or `"b 1010011100111011 message4.txt".`

Input example:

```
CRYPTOGRAPHY HAS BEEN A MAJOR ELEMENT IN FICTION OVER THE LAST
TWO HUNDRED YEARS. DETECTIVE FICTION HAS BEEN A PARTICULARLY
FERTILE GROUND FOR THIS PLOT ELEMENT STARTING WITH EDGAR ALLAN
POES THE GOLD BUG.
```

Output example:

```
Encryption KEY:  1010011100111011

Input Message:

CRYPTOGRAPHY HAS BEEN A MAJOR ELEMENT IN FICTION OVER THE LAST
TWO HUNDRED YEARS. DETECTIVE FICTION HAS BEEN A PARTICULARLY
FERTILE GROUND FOR THIS PLOT ELEMENT STARTING WITH EDGAR ALLAN
POES THE GOLD BUG.

Output Message:

0xfa 0x64 0x79 0xe8 0x25 0xbe 0xf0 0x34 0xd4 0x99
```

```
0xee 0x46 0x83 0x30 0x88 0xd5 0x26 0x71 0x8c 0x20
0x8f 0x00 0x09 0x57 0x02 0xed 0x2d 0x61 0x2f 0x01
0x3d 0xfb 0x71 0xac 0xeb 0x86 0x3a 0x6b 0x01 0xa7
0x0c 0x27 0x0b 0x8d 0x5f 0xac 0x23 0x31 0x8d 0x60
0x5a 0x6c 0xc6 0xc6 0xae 0x9e 0xd6 0x7b 0xf6 0xc4
0x36 0x7b 0x01 0xad 0xf4 0x94 0x58 0xdc 0x35 0x8b
0x66 0x7a 0x15 0xb3 0x0b 0x8d 0xeb 0x86 0x3a 0x6b
0x01 0xa7 0xac 0x2e 0x64 0x9a 0x22 0xe1 0x76 0x7c
0xd4 0x99 0xf4 0x94 0x15 0xb3 0x2a 0x61 0x8d 0x60
0x32 0xe9 0x41 0xaf 0xf5 0x84 0x15 0xb3 0x2d 0x61
0xf0 0x34 0x23 0x11 0x3c 0x2b 0x0b 0x87 0x5f 0xac
0xe3 0x36 0x7a 0x68 0x0d 0x67 0x05 0xbd 0x2d 0x61
0x2f 0x01 0x3d 0xfb 0x5a 0x6c 0xf4 0x94 0x15 0xb3
0x4c 0x23 0x14 0x43 0xb5 0xb7 0x36 0x7b 0x84 0x40
0xf2 0xea 0x5d 0x68 0x70 0x1c 0x27 0xce 0x65 0x8a
0xb5 0xb7 0x46 0x73 0x5e 0x98 0xf5 0xb4
```

Program's Design.

Important Variables.

The program uses String global variables calles input and output that store the message given to the program and the results of the process. It also has a String called inputProcessed that stores the input message without white spaces and non-alphabetical characters. It also handles three String arrays (state, sbox and expandedKey) that help representing the state of the encryption, the look-up table for substitution and the expanded key.

Methods for number/string conversions.

convertByteToBinary(byte)

This method takes a byte representing a character and returns a string representing the binary number of the character.

convertBinaryStringToInteger(String)

This method takes a string representing a binary number and returns a decimal representation of it.

stateToWord()

This method returns a string of a 16-bit number representing the state of the encryption.

Methods for S AES encryption.

addRoundKey(String)

This method adds the round key to the state. It uses a helper method called addWord(String,String) that takes two binary-number strings and XOR them together.

substitution()

This method takes every nibble of the state and substitutes them using the look-up table. It uses a the helper method subNibble(String) that takes a nibble from the state and it returns its substitution.

mixcolumns()

This method makes the bit-wise additions necessary to mix the column of the state.

encipher()

This method takes every pair of characters of the input message and enciphers them using a helper method called encipherBlock(byte, byte), which takes two bytes representing the characters.

main().

This method reads the input from a text file and uses the method gatherInput(String) in order to store the contents of the file and process it for later encryption. Furthermore, it is also necessary to mention that this method deals with possible three invalid arguments from the command line by showing a message about the error and ending the program.