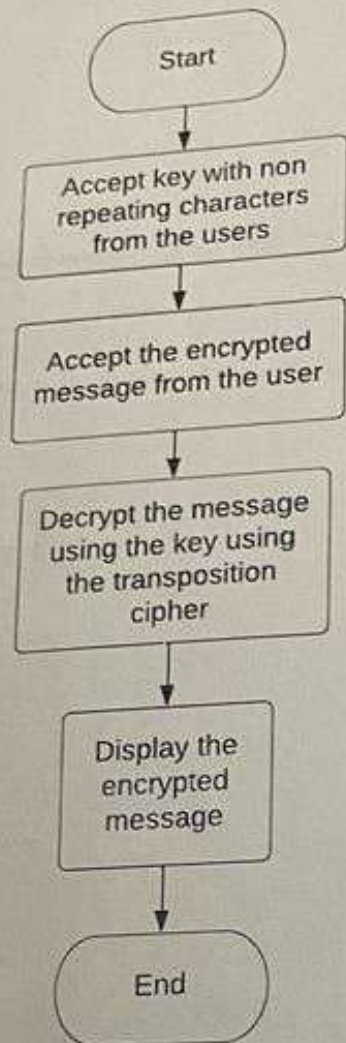


- 1) Start
- 2) Store a password in the code.
- 3) Accept a password from the user using getch()
- 4) Print "*" for all the user input characters.
- 5) Compare the user input string and password.
- 6) If true, then display "Correct password".
- 7) Else display "Wrong Password".
- 8) End



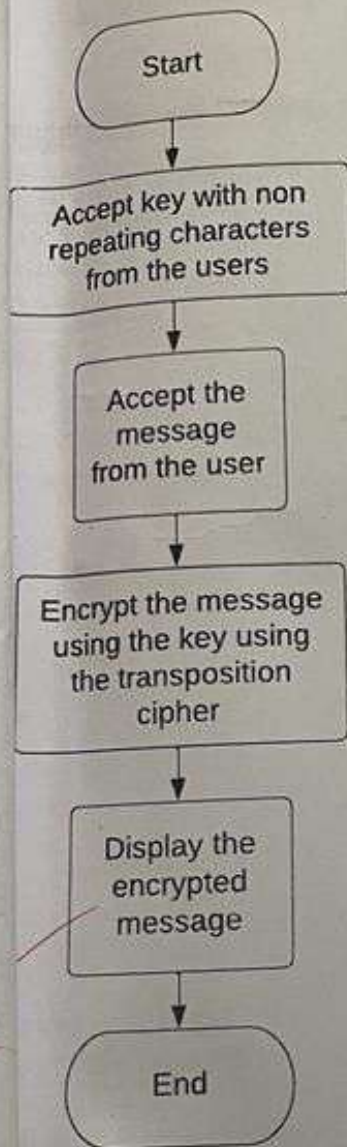
Algorithm:

Encryption:

1. Start
2. Accept key with non-repeating characters from the user.
3. Extract key length.
4. Accept message from the user.
5. Use key length to determine size of matrix with respect to the message.
6. Add random characters to make a full matrix if required.
7. Sort the key in ascending order.
8. Create encrypted message with the sorted key.
9. Display the encrypted message.

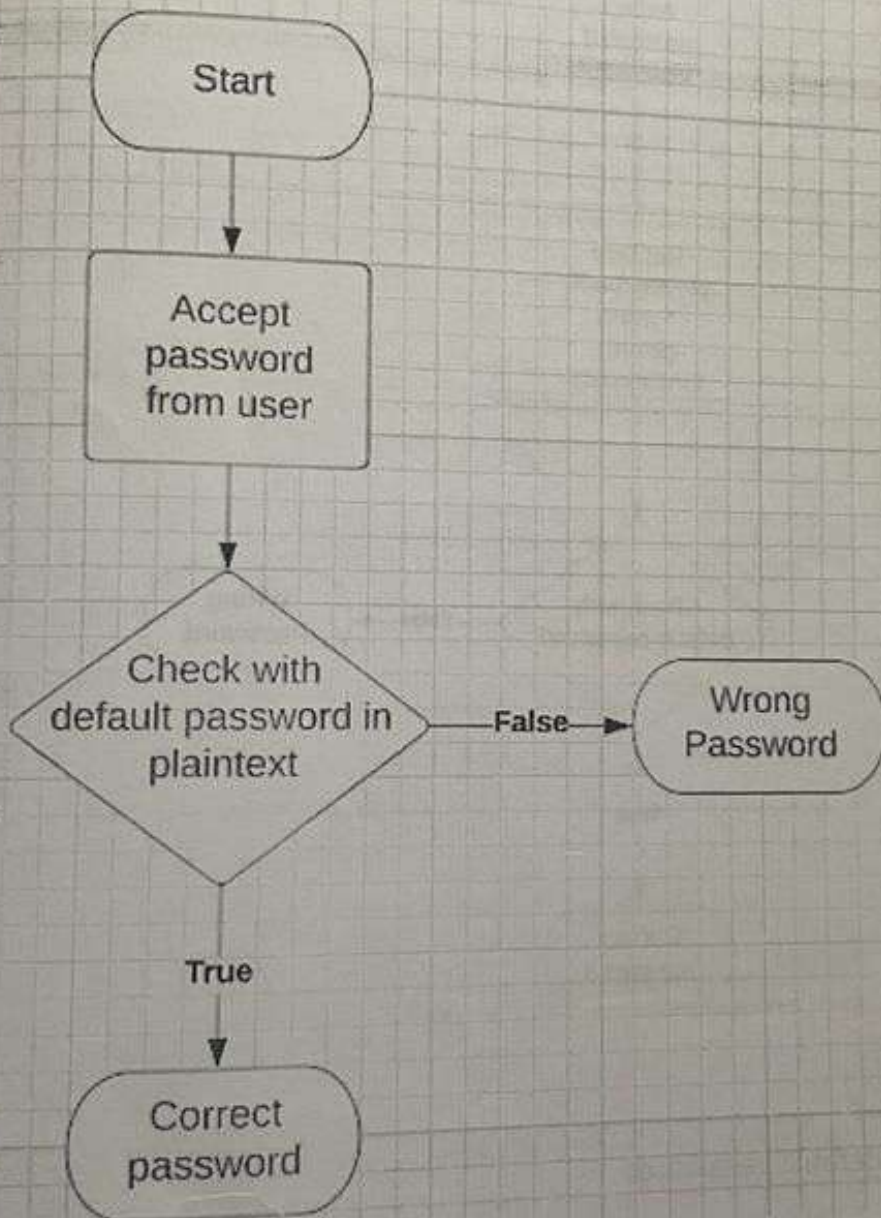
Experiment No. 2

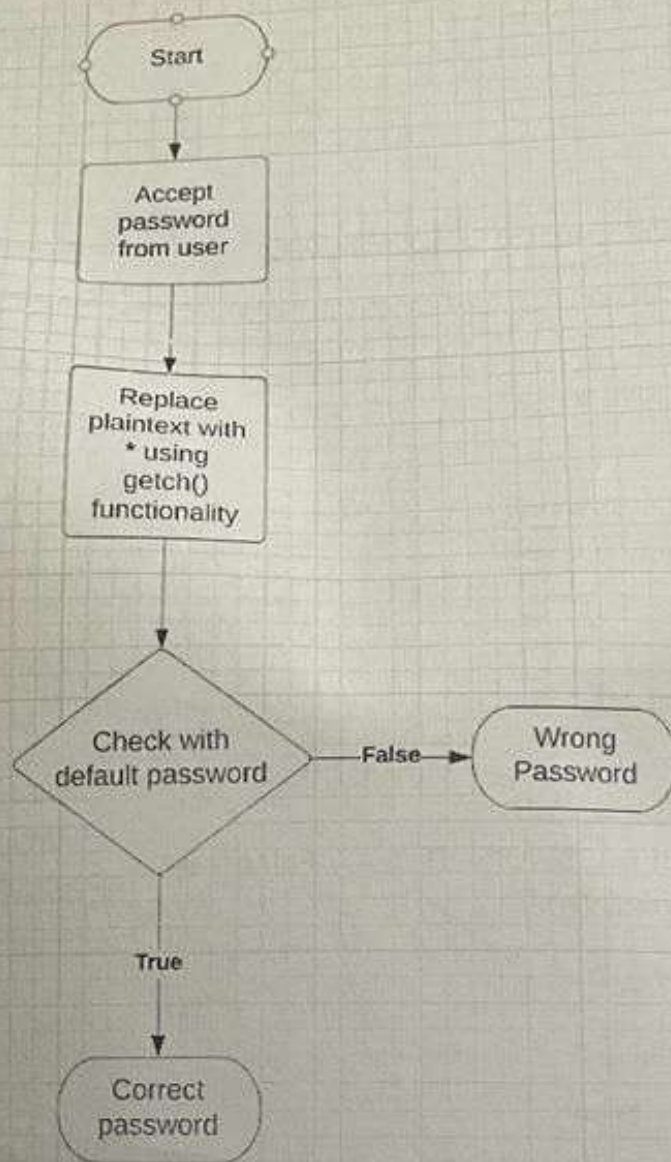
Flowchart:



Experiment No. 1

Flowchart:





Algorithm:

For plaintext + default password:

- 1) Start
- 2) Store a default password like "admin" or "test".
- 3) Accept a password as a string from user.
- 4) Compare the default password and the user input.
- 5) If true display correct password
- 6) Else display wrong password.
- 7) End

For Printed password:

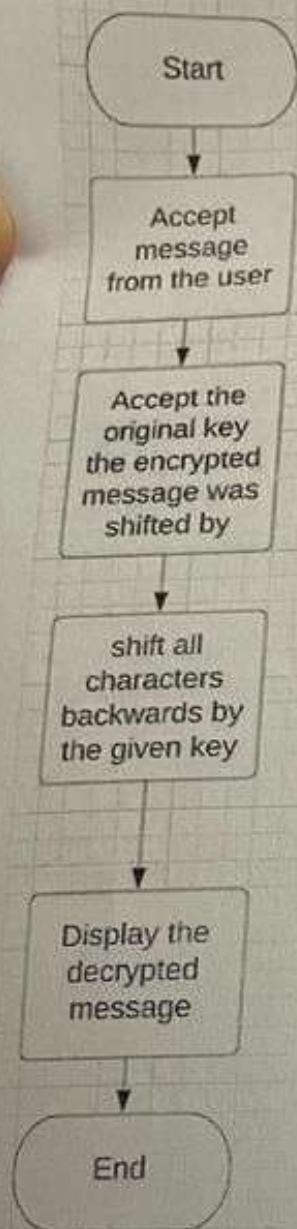
10.End.

Decryption:

1. Start
2. Accept key used to encrypt the data.
3. Accept the encrypted message from the user.
4. Decrypt the encryption using the key.
5. Display the decrypted message.
6. end

he user.

respect to the message.
required.



1. Start
2. Accept
3. Accept
4. Shift b
5. Displa
6. End

Algorithm:

Encyption:

1. Start
2. Accept the message from the user.
3. Accept the integer key from the user.
4. Shift forward all the characters in the message by the key.
5. Display the encrypted message.
6. End

Decryption:

1. Start
2. Accept the encrypted message form the user.
3. Accept the integer key used for encryption.
4. Shift backwards all the characters by the key amount.
5. Display the message.
6. End

Experiment No. 3

Flowchart:

