EXP NO: 1c

DATE: 10/02/2024

## RAIL FENCE CIPHER

AIM:

To write a python program implementing rail fence cipher algorithm

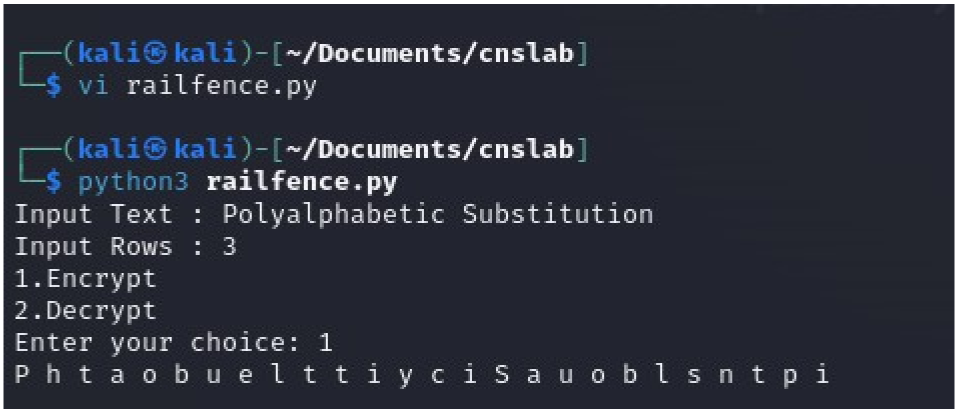
ALGORITHM:

1. Get the plain text from the user
2. Set the key as 2 by default.
3. Arrange the plaintext in two rows in a zig-zag manner.
4. Derive the cipher text by adding the first row of arrangement with the second row of arrangement.
5. Get the original text by using the cipher text and arranging it in zigzag manner and repeat the same process.

PROGRAM:

```
def main():
    text = input('Input Text : ')
    rows = int(input('Input Rows : '))
    text = text.replace(' ',")

    while True:
            chc = input('1.Encrypt\n2.Decrypt\nEnter your choice: ')
            if chc in ['0','1']:
                break
                print('Choose 0 / 1')

    #print(len(text))
    if int(chc):
            arr = [[ ' ' for y in range(len(text))] for x in range(rows)]
            #[ print(row) for row in arr ]

            dir_down = None
            row, col = 0 , 0
            for i in range(len(text)):
                if row == 0: dir_down = True
                if row == rows - 1: dir_down = False

                arr[row][col] = '*'
                col += 1

                if dir_down: row += 1
                else: row -= 1

            #print('\n\n')
            #[ print(row) for row in arr ]
            count = 0
            for row in arr:
                for i in range(len(row)):
                    if row[i] == '*':
                        row[i] = text[count]
                        count += 1

            #print('\n\n')
            #[ print(row) for row in arr ]

            result = []
            row, col = 0, 0
            for i in range(len(text)):

                if row == 0: dir_down = True
                if row == rows-1: dir_down = False

                if (arr[row][col] != '*'):
                    result.append(arr[row][col])
                    col += 1

                if dir_down: row += 1
                else: row -= 1

            print(" ".join(result).strip())
        else:
            arr = [ [] for x in range(rows)]
            #print(arr)
            count = 0
            finish = False

            while True:
                for j in range(0,rows-1):
                    arr[j].append(text[count])
                    count += 1

                    if count >= len(text):
                        finish = True
                        break

                if finish :
                    break
                for k in range(rows - 1 ,0,-1):
                    arr[k].append(text[count])
                    count += 1

                    if count >= len(text):
                        finish = True
                        break

                if finish :
                    break
            print(arr)

main()
```

OUTPUT:



```
┌──(kali㉿kali)-[~/Documents/cnslab]
└─$ vi railfence.py

┌──(kali㉿kali)-[~/Documents/cnslab]
└─$ python3 railfence.py
Input Text : Polyalphabetic Substitution
Input Rows : 3
1.Encrypt
2.Decrypt
Enter your choice: 1
P h t a o b u e l t t i y c i S a u o b l s n t p i
```

RESULT:

Thus, a python program has been implemented to demonstrate Rail Fence Cipher.