

**DATE:****RAILFENCE CIPHER****AIM:**

To implement encryption and decryption in railfence cipher technique

**ALGORITHM:**

- The user must enter the plain text to be encrypted
- The user enters the depth as the key
- Create the matrix based on length of plain text and depth
- Place the each character in a diagonal way one column after another
- Print the text as cipher text row wise
- For decryption do the process in opposite way

**PROGRAM CODE:**

```
def encryptRailFence(text, key):
    rail = [['\n' for i in range(len(text))] for j in range(key)]
    dir_down = False
    row, col = 0, 0

    for i in range(len(text)):
        if (row == 0) or (row == key - 1):
            dir_down = not dir_down
        rail[row][col] = text[i]
        col += 1
        if dir_down:
            row += 1
        else:
            row -= 1
    result = []
    for i in range(key):
        for j in range(len(text)):
            if rail[i][j] != '\n':
                result.append(rail[i][j])
    return("".join(result))

def decryptRailFence(cipher, key):
    rail = [['\n' for i in range(len(cipher))] for j in range(key)]
    dir_down = None
    row, col = 0, 0
    for i in range(len(cipher)):
        if row == 0:
            dir_down = True
        if row == key - 1:
            dir_down = False
        rail[row][col] = '*'
        col += 1
        if dir_down:
            row += 1
        else:
            row -= 1
```

```

        else:
            row -= 1
index = 0
for i in range(key):
    for j in range(len(cipher)):
        if ((rail[i][j] == '*') and
            (index < len(cipher))):
            rail[i][j] = cipher[index]
            index += 1
result = []
row, col = 0, 0
for i in range(len(cipher)):
    if row == 0:
        dir_down = True
    if row == key-1:
        dir_down = False
    if (rail[row][col] != '*'):
        result.append(rail[row][col])
        col += 1
    if dir_down:
        row += 1
    else:
        row -= 1
return("".join(result))

```

```

plain_text=input("Enter the plain text to be encrypted: \n")
pdepth=int(input("Enter the depth of encryption: \n"))
cipher_text=input("Enter the cipher text to be decrypted: \n")
cdepth=int(input("Enter the depth of decryption: \n"))
print("Encrypted Text : " + encryptRailFence(plain_text, pdepth))
print("Decrypted Text : "+decryptRailFence(cipher_text, cdepth ))

```

## OUTPUT:

```

Enter the plain text to be encrypted:
ASTRONAUT
Enter the depth of encryption:
4
Enter the cipher text to be decrypted:
BSLSI
Enter the depth of decryption:
3
Encrypted Text : AASNUTOTR
Decrypted Text : BLISS

```

## RESULT: