

# Cryptography---19CS412-classical-techqniques

Name:SARWESHVARAN A

Regno:212223230198

## Rail Fence Cipher

Rail Fence Cipher using with different key values

### AIM:

To develop a simple C program to implement Rail Fence Cipher.

### DESIGN STEPS:

#### Step 1:

Design of Rail Fence Cipher algorithm

#### Step 2:

Implementation using C or pyhton code

#### Step 3:

Testing algorithm with different key values. ALGORITHM DESCRIPTION: In the rail fence cipher, the plaintext is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when we reach the bottom rail. When we reach the top rail, the message is written downwards again until the whole plaintext is written out. The message is then read off in rows.

### PROGRAM:

```
#include <stdio.h>
#include <string.h>

int main() {
    int i, j, k, l;
    char a[20], c[20], d[20];

    printf("\n\t\t RAIL FENCE TECHNIQUE");
    printf("\n\nEnter the input string : ");
    fgets(a, sizeof(a), stdin);
    a[strcspn(a, "\n")] = '\0'; // Remove newline

    l = strlen(a);

    // Encryption
    for (i = 0, j = 0; i < l; i++) {
        if (i % 2 == 0)
            c[j++] = a[i];
    }
    for (i = 0; i < l; i++) {
        if (i % 2 == 1)
            c[j++] = a[i];
    }
    c[j] = '\0';
```



```

printf("\nCipher text after applying rail fence: %s", c);

// Decryption
if (l % 2 == 0)
    k = l / 2;
else
    k = (l / 2) + 1;

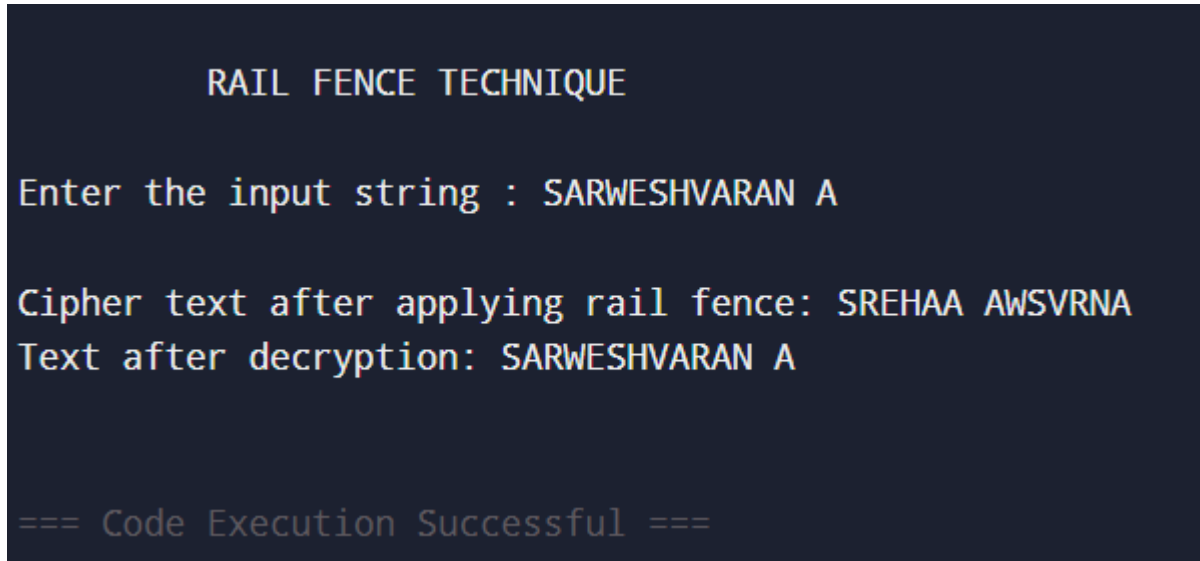
for (i = 0, j = 0; i < k; i++) {
    d[j] = c[i];
    j += 2;
}
for (i = k, j = 1; i < l; i++) {
    d[j] = c[i];
    j += 2;
}
d[l] = '\0';

printf("\nText after decryption: %s\n", d);

return 0;
}

```

## OUTPUT:



```

RAIL FENCE TECHNIQUE

Enter the input string : SARWESHVARAN A

Cipher text after applying rail fence: SREHAA AWSVRNA
Text after decryption: SARWESHVARAN A

=== Code Execution Successful ===

```

## RESULT:

The program is executed successfully