**Theory:**

The plaintext is written downwards and diagonally on successive “rails”, then moving up when we reach the bottom rail. When we reach the top rail, the message is written downwards again until whole plaintext is written out.

**Programming Language: C**

**IDE: DEV C++**

**Code :**

#include <stdio.h>

#include <string.h>

void encryptRailFence(char \*plaintext, int rails) {

int len = strlen(plaintext);

char ciphertext[len];

int r = 0, i, j, k = 0;

for (i = 0; i < rails; i++) {

for (j = i; j < len; j += rails) {

ciphertext[k++] = plaintext[j];

if (i != 0 && i != rails - 1 && j + (rails - i - 1) \* 2 < len)

ciphertext[k++] = plaintext[j + (rails - i - 1) \* 2];

}

}

ciphertext[k] = '\0';

printf("Encrypted Text: %s\n", ciphertext);

}

void decryptRailFence(char \*ciphertext, int rails) {

int len = strlen(ciphertext);

char plaintext[len];

int r = 0, i, j, k = 0;

for (i = 0; i < rails; i++) {

for (j = i; j < len; j += rails) {

plaintext[j] = ciphertext[k++];

if (i != 0 && i != rails - 1 && j + (rails - i - 1) \* 2 < len)

plaintext[j + (rails - i - 1) \* 2] = ciphertext[k++];

}

}

plaintext[len] = '\0';

printf("Decrypted Text: %s\n", plaintext);

}

int main() {

char plaintext[100];

int rails;

printf("Enter the plaintext: ");

fgets(plaintext, sizeof(plaintext), stdin);

plaintext[strcspn(plaintext, "\n")] = '\0';

printf("Enter the number of rails: ");

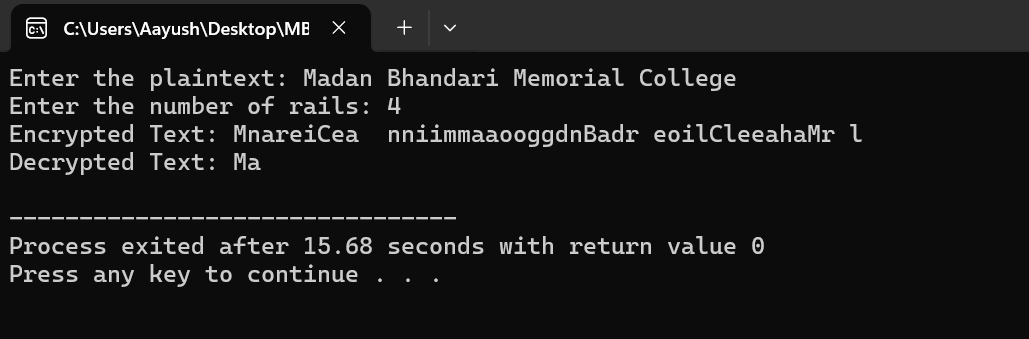
scanf("%d", &rails);

encryptRailFence(plaintext, rails);

decryptRailFence(plaintext, rails);

return 0;

}

****

**Output**