**EXPERIMENT- 05: RAIL FENCE TECHNIQUE**

**PROGRAM:**

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

void encrypt(char *text, int key) {
   int len = strlen(text);
   char rail[key][len];
   for (int i = 0; i < key; i++)
      for (int j = 0; j < len; j++)
         rail[i][j] = '\n';

   int row = 0, col = 0;
   int dir_down = 0;

   for (int i = 0; i < len; i++) {
      if (row == 0)
         dir_down = 1;
      if (row == key - 1)
         dir_down = 0;

      rail[row][col++] = text[i];

      dir_down ? row++ : row--;
   }

   for (int i = 0; i < key; i++)
      for (int j = 0; j < len; j++)
         if (rail[i][j] != '\n')
            printf("%c", rail[i][j]);
}

void decrypt(char *cipher, int key) {
   int len = strlen(cipher);
   char rail[key][len];
   for (int i = 0; i < key; i++)
      for (int j = 0; j < len; j++)
         rail[i][j] = '\n';

   int row = 0, col = 0;
   int dir_down = 0;

   for (int i = 0; i < len; i++) {
      if (row == 0)
         dir_down = 1;
      if (row == key - 1)
         dir_down = 0;

      rail[row][col++] = '*';

      dir_down ? row++ : row--;
   }

   int index = 0;
   for (int i = 0; i < key; i++)
      for (int j = 0; j < len; j++)
         if ((rail[i][j] == '*') && (index < len))
            rail[i][j] = cipher[index++];
```

```
    row = 0, col = 0;
    dir_down = 0;

    for (int i = 0; i < len; i++) {
        if (row == 0)
            dir_down = 1;
        if (row == key - 1)
            dir_down = 0;

        if (rail[row][col] != '\n')
            printf("%c", rail[row][col++]);

        dir_down ? row++ : row--;
    }
}

int main() {
    int choice, key;
    char text[100];

    printf("Select 1 for Encryption or 2 for Decryption: ");
    scanf("%d", &choice);
    printf("Enter the key: ");
    scanf("%d", &key);
    printf("Enter the text: ");
    scanf("%s", text);

    if (choice == 1) {
        encrypt(text, key);
    } else if (choice == 2) {
        decrypt(text, key);
    } else {
        printf("Invalid choice.\n");
    }

    return 0;
}
```

output: