ENCRYPTION OF PLAIN TEXT USING RAILFENCE

AIM:

To implement a C program to perform Railfence encryption

ALGORITHM:

I)In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

II)In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.

III)When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.

Iv)After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

Generated by CamScanner

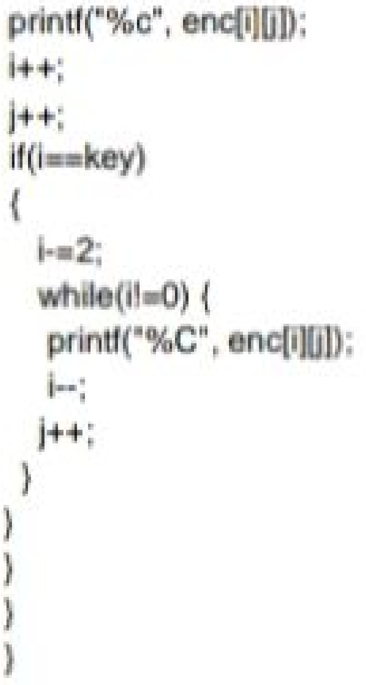
PROGRAM:

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

void main(){
   char pt [100];
   char enc[100] [100]={""};
   int key;
   printf("\nEnter the plain text :");
   scanf("%s", pt);
   printf("\nEnter the key :");
   scanf("%d", &key);
   int i=0,j=0,k=0;
   while(j<=strlen(pt))
   {
     enc[i][j] = pt[j];
     i++;
     j++;
   if (i==key)
   {


     i-=2;
     while(i!=0){
         enc[i][j] = pt[j];
         i--;
         j++;
     }
   }
   }
printf("\nEncryption :");
for (i=0; i<key; i++)

for (j=0; j<strlen(pt); j++)
if (enc[i][j]!= "")
printf("%c", enc[i][j]);
i=0;
j=0;

printf("\nDecryption:");
while(j<=strlen(pt))
{
 printf("%c", enc[i][j]);
 i++;
 j++;
 if(i==key)
 {
    i-=2;
```

Generated by CamScanner

```
     while(i!=0) {
      printf("%C", enc[i][j]);
      i--;
     j++;
   }
  }
  }
  #include<stdio.h>
  #include<string.h>
  #include<ctype.h>

  void main(){
    char pt [100];
    char enc[100] [100]={""};
    int key;
    printf("\nEnter the plain text :");
    scanf("%s", pt);
    printf("\nEnter the key :");
    scanf("%d", &key);
    int i=0,j=0,k=0;
    while(j<=strlen(pt))
    {
      enc[i][j] = pt[j];
      i++;
      j++;
    if (i==key)
    {


      i-=2;
      while(i!=0){
          enc[i][j] = pt[j];
          i--;
          j++;
      }
    }
    }
  printf("\nEncryption :");
  for (i=0; i<key; i++)

  for (j=0; j<strlen(pt); j++)
  if (enc[i][j]!= "")
  printf("%c", enc[i][j]);
  i=0;
  j=0;

  printf("\nDecryption:");
  while(j<=strlen(pt))
  {
```

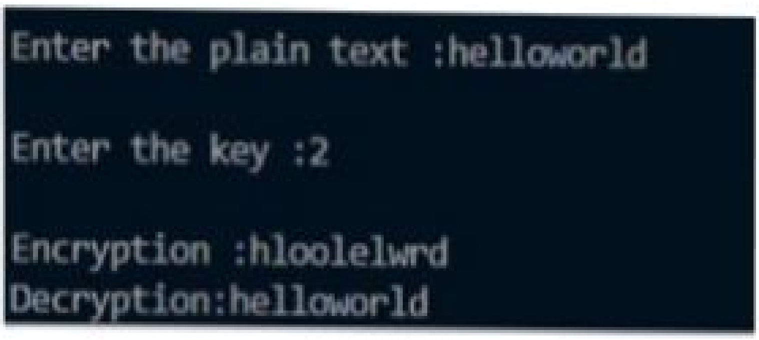
Generated by CamScanner



```
printf("%c", enc[i][j]);
i++;
j++;
if(i==key)
{
   i-=2;
   while(i!=0) {
    printf("%C", enc[i][j]);
    i--;
   j++;
 }
}
}
}
}
```

Generated by CamScanner

OUTPUT:



```
Enter the plain text :helloworld

Enter the key :2

Encryption :hloolelwrd
Decryption:helloworld
```

RESULT:

Thus the c program to implement Railfence encryption executed successfully.

Generated by CamScanner