/*C Program to implement RSA encryption and decryption

Input : 1. Two prime numbers

2. Message to be encrypted

Output : Encrypted message

***/**

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>

long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;
char msg[100];
int prime(long int);
void ce();
long int cd(long int);
void encrypt();
void decrypt();
void main()
{
//clrscr();
printf("\n Enter a prime number.\n");
scanf("%d",&p);
flag=prime(p);
if(flag==0)
{
   printf("\n No. That's not a prime number!\n");
   getch();
   exit(1);
}
```

```
printf("\n Enter another prime number.\n");
scanf("%d",&q);
flag=prime(q);
if(flag==0||p==q)
{
    printf("\n Not a prime number!\n");
    getch();
    exit(1);
}
printf("\n Enter the message you want to encrypt.\n");
fflush(stdin);
scanf("%s",msg);
for(i=0;msg[i]!=NULL;i++)
m[i]=msg[i];
n=p*q;
t=(p-1)*(q-1);
ce();
printf("\n Possible values of 'd' and 'e' are\n");
for(i=0;i<j-1;i++)
printf("\n%ld\t%ld",e[i],d[i]);
encrypt();
decrypt();
getch();
}
int prime(long int pr)
{
int i;
j=sqrt(pr);
for(i=2;i<=j;i++)
{
    if(pr%i==0)
```

```
    return 0;
}
return 1;
}
void ce()
{
int k;
k=0;
for(i=2;i<t;i++)
{
    if(t%i==0)
    continue;
    flag=prime(i);
    if(flag==1&&i!=p&&i!=q)
    {
        e[k]=i;
        flag=cd(e[k]);
        if(flag>0)
        {
            d[k]=flag;
            k++;
        }
        if(k==99)
        break;
    }
}
}
long int cd(long int x)
{
long int k=1;
while(1)
```

```
{
   k=k+t;
   if(k%x==0)
   return(k/x);
}
}
void encrypt()
{
long int pt,ct,key=e[0],k,len;
i=0;
len=strlen(msg);
while(i!=len)
{
   pt=m[i];
   pt=pt-96;
   k=1;
   for(j=0;j<key;j++)
   {
      k=k*pt;
      k=k%n;
   }
   temp[i]=k;
   ct=k+96;
   en[i]=ct;
   i++;
}
en[i]=-1;
printf("\n\n The encrypted message is\n ");
for(i=0;en[i]!=-1;i++)
printf("%c",en[i]);
}
```

```
void decrypt()
{
long int pt,ct,key=d[0],k;
i=0;
while(en[i]!=-1)
{
   ct=temp[i];
   k=1;
   for(j=0;j<key;j++)
   {
      k=k*ct;
      k=k%n;
   }
   pt=k+96;
   m[i]=pt;
   i++;
}
m[i]=-1;
printf("\n\n The decrypted message is\n ");
for(i=0;m[i]!=-1;i++)
printf("%c",m[i]);
printf("\n\n ");
}
```

Sample Input and Output:

```
Enter a prime number.
7

Enter another prime number.
11

Enter the message you want to encrypt.
BeHuman

Possible values of 'd' and 'e' are

13      37
17      53
19      19
23      47
29      29
31      31

The encrypted message is
↑z,uëaæ

The decrypted message is
BeHuman

Press any key to continue...
```