

RSA algorithm to encrypt and decrypt the data.

```
import java.math.BigInteger;
import java.util.*;
class rsa
{
public static void main(String args[])
{
    Scanner ip=new Scanner(System.in);
    int p,q,n,e=1,j;
    int d=1,i1;
    int t1,t2;
    int pt[]= new int[10];
    int ct[]= new int[10];
    int rt[]= new int[10];
    int temp[]= new int[10];
    String i=new String();

    System.out.println("Enter the two prime numbers:");
    p=ip.nextInt();
    q=ip.nextInt();

    System.out.println("Enter the message to be sent");
    i=ip.next();
    i1=i.length();
    n=p*q;
    t1=p-1;
    t2=q-1;
```

RSA algorithm to encrypt and decrypt the data.

```
System.out.println("\n-----");
System.out.println("Sender Side:");
while((t1*t2)%e==0)
{
    e++;
}
System.out.println("Public Key(e)= "+e);
System.out.println("-----");
for(j=0;j<i1;j++)
{
    pt[j]=(i.charAt(j))-96;
// System.out.println("Plain Text= "+pt[j]);
    ct[j]=((int)Math.pow(pt[j],e))%n;
    System.out.println("Cipher Text= "+ct[j]);
}
System.out.println("\nTransmitted Message:");
for(j=0;j<i1;j++)
{
    temp[j]=ct[j]+96;
    System.out.print((char)temp[j]);
}

System.out.println("\n\n-----");
System.out.println("Receiver Side:");
while((d*e)%(t1*t2)!=1)
{
    d++;
}
```

RSA algorithm to encrypt and decrypt the data.

```
System.out.println("Private Key(d)= "+d);
System.out.println("-----");

for(j=0;j<i1;j++)
{
    //System.out.println("cipher Text= "+ct[j]);
    BigInteger very_big_no = BigInteger.valueOf(ct[j]);
    very_big_no = very_big_no.pow(d);
    very_big_no = very_big_no.mod(BigInteger.valueOf(n));
    rt[j] = very_big_no.intValue();
    System.out.println("Plain Text= "+rt[j]);
}

System.out.println("\n-----");
System.out.println("Decrypted Message:");
for(j=0;j<i1;j++)
{
    rt[j]=rt[j]+96;
    System.out.print((char)rt[j]);
}
System.out.println("\n-----");
ip.close();
}
}
```

RSA algorithm to encrypt and decrypt the data.

Output:

```
java rsa
Enter the two prime numbers:
5
11
Enter the message to be sent
global
```

```
-----
Sender Side:
Public Key(e)= 3
```

```
-----
Cipher Text= 13
Cipher Text= 23
Cipher Text= 20
Cipher Text= 8
Cipher Text= 1
Cipher Text= 23
```

```
Transmitted Message:
mwthaw
```

```
-----
Receiver Side:
Private Key(d)= 27
```

```
-----
Plain Text= 7
Plain Text= 12
Plain Text= 15
Plain Text= 2
Plain Text= 1
Plain Text= 12
```

```
-----
Decrypted Message:
global
-----
```