Practical-13

Practical Statement : Write a program to implement two Digital Signature Algorithms: DSA and Elgamal.

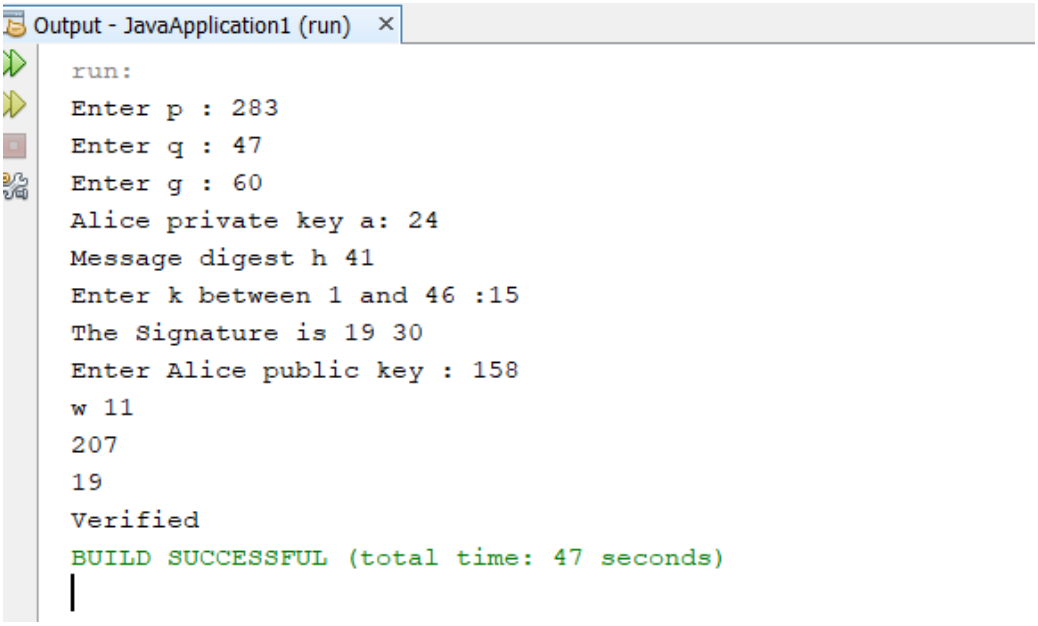
Code for Digital Signature Algorithm :

```
import java.util.Scanner;
public class DSA {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter p : ");
        int p = scan.nextInt();
        System.out.print("Enter q : ");
        int q = scan.nextInt();
        System.out.print("Enter g : ");
        int g = scan.nextInt();
        System.out.print("Alice private key a: ");
        int a = scan.nextInt();
        System.out.print("Message digest h ");
        int h = scan.nextInt();
        System.out.print("Enter k between 1 and " + (q-1) + " :");
        int k = scan.nextInt();
        int x = g;
        for(int i=2;i<=k;i++) {
            x = (x*g)%p;
        }
        int r = x%q;
        int s=0;
        if(r!=0) {
            int kinverse=EUA(q,k);
            s  = kinverse*(h + (a*r))%q;
            if(s!=0){
                System.out.println("The Signature is " + r + " " + s);
            }
        }
      System.out.print("Enter Alice public key : ");
```

```
    int A = scan.nextInt();
    if(r<q && s<q) {
        int w = EUA(q,s);
        System.out.println("w " + w);
        int u1 = (h*w)%q;
        int u2 = (r*w)%q;
        int bx1 = g;
        for(int i=2;i<=u1;i++)
            bx1 = (bx1*g)%p;
        int bx2 = A;
        for(int i=2;i<=u2;i++)
            bx2 = ((bx2%p)*(A%p))%p;
        int bx = (bx1*bx2)%p;
        System.out.println(bx);
        int v = bx%q;
        System.out.println(v);
        if(v==r) {
            System.out.println("Verified");
        }
    }
}
public static int EUA(int fieN,int e) {
    int a1 = 1;
    int a2 = 0;
    int a3 = fieN;
    int b1 = 0;
    int b2 = 1;
    int b3 = e;
    int q = 0;
    int ans = b2;
    while(b3!=0 && b3!=1) {
        q = a3/b3;
        int t1 = a1;
        int t2 = a2;
        int t3 = a3;
```

```
            a1 = b1;
            a2 = b2;
            a3 = b3;
            b1 = t1 - q*b1;
            b2 = t2 - q*b2;
            b3 = t3 - q*b3;
            if(b3==1) {
                if(b2 < 0) {
                    b2 = fieN + b2;
                }
                ans = b2;
            }
            else{
                ans = 0;
            }
        }
        return ans;
    }
}
```

Output :



Code for Elgamal Digital Signature Algorithm :

```
import java.util.Scanner;
public class Elgamal{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter q : ");
        int q = scan.nextInt();
        System.out.print("Enter alpha : ");
        int a = scan.nextInt();
        System.out.print("Enter random number x between 1 and " + (q-1) + " :");
        int xa = scan.nextInt();
        int ya = a;
        for(int i=2;i<=xa;i++)
            ya = (ya*a)%q;
        System.out.println("public key is " + xa);
        System.out.println("private key is " + q + " " + a + " " + ya);
        System.out.print("Enter hash value m : ");
        int m = scan.nextInt();
        int k=0;
        for(int i=2;i<=q-1;i++){
            if(gcd(i,q-1)==1) {
                k=i;
                break;
            }
        }
        int s1 = a;
        for(int i=2;i<=k;i++)
            s1 = (s1*a)%q;
        int kinvers = EUA(q-1,k);
        int s2 = kinvers*(m-(xa*s1))%(q-1);
        while(s2<0)
            s2=s2+(q-1);
        s2=s2%(q-1);
        //bob verification
```

```
    int v1 = a;
    for(int i=2;i<=m;i++)
        v1=(v1*a)%q;
    int v21 = ya;
    for(int i=2;i<=s1;i++)
        v21 = (v21*ya)%q;
    int v22 = s1;
    for(int i=2;i<=4;i++)
        v21 = (v21*s1)%q;
    int v2 = (v21*v22)%q;
    if(v1==v2)
        System.out.println("verified");
    else
        System.out.println("Not verified");
}
static int gcd(int a,int b) {
    if (b == 0) {
        return a;
    }
    return gcd(b, a % b);
}
public static int EUA(int fieN,int e) {
    int a1 = 1;
    int a2 = 0;
    int a3 = fieN;
    int b1 = 0;
    int b2 = 1;
    int b3 = e;
    int q = 0;
    int ans = b2;
    while(b3!=0 && b3!=1) {
        q = a3/b3;
        int t1 = a1;
        int t2 = a2;
```

```
        int t3 = a3;
        a1 = b1;
        a2 = b2;
        a3 = b3;
        b1 = t1 - q*b1;
        b2 = t2 - q*b2;
        b3 = t3 - q*b3;
        if(b3==1) {
            if(b2 < 0) {
                b2 = fieN + b2;
            }
            ans = b2;
        }
        else{
            ans = 0;
        }
    }
    return ans;
  }
}
```

Output :