**Program:**

**server.cpp -**

```
#include <iostream>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#define SA struct sockaddr
#define typeL unsigned long
using namespace std;

typeL powermod(typeL a, typeL b, typeL  q)
{
	typeL res=1;
	for(typeL i=0;i<b;i++)
		res=(res*a)%q;
	return res;
}

typeL inverse(typeL k , typeL x)
{
	typeL d=x, r=k, t, q[100],p[100];
	int i=0;
	do{
		t=d;
		q[i]=d/r;
		d=r;
		r=t%r;
		if(i==0 || i==1) p[i]=i;
		else p[i]=(p[i-2]-p[i-1]*q[i-2])%x;
		i++;
	}while(r!=0);
	p[i]=(p[i-2]-p[i-1]*q[i-2])%x;
	return p[i];
}

typeL H(typeL M)
{
	return ( M^1234 ); //hash key=1234
}

typeL f1(typeL M,typeL k,typeL x,typeL r,typeL q)
{
	return ( inverse(k,q) * ( H(M) + x*r ) )%q;
}

typeL f2(typeL k, typeL p, typeL q, typeL g)
{
	return powermod(g,k,p)%q;
}
```

```
int main()
{
	srand(time(NULL));
	int port;
	char addr[100]={'\0'};
	cout<<"Address  : "; scanf("%s",addr);
	cout<<"Port     : "; cin>>port;

	typeL p,q,r,s,k,g,M,h,x,y,hashval;

	cout<<"p = "; cin>>p;
	cout<<"q = "; cin>>q;
	cout<<"M = "; cin>>M;

	hashval=H(M);
	h=rand()%(p-3)+2;
	g=powermod(h,(p-1)/q,p);

	x=rand()%(q-2)+1;     //User private key
	y=powermod(g,x,p);  //User public key

	k=rand()%(q-2)+1;

	//Signing
	r=f2(k,p,q,g);
	s=f1(M,k,x,r,q);

	// ****Connection
	struct sockaddr_in server={AF_INET, htons(port), inet_addr(addr)}, client;
	int sockfd = socket(AF_INET, SOCK_STREAM,0);
	bind(sockfd, (SA*)&server, sizeof(server));
	listen(sockfd,1);
	socklen_t len=sizeof(client);
	int connfd = accept(sockfd,(SA*)&client,&len);
	// ****Connection Established

	send(connfd, &hashval, sizeof(hashval), 0);
	send(connfd, &r, sizeof(r), 0);
	send(connfd, &s, sizeof(s), 0);
	send(connfd, &g, sizeof(g), 0);
	send(connfd, &y, sizeof(y), 0);

	cout<<"Packet sent with values"<<endl;
	cout<<"Hash : "<<hashval<<endl;
	cout<<"R    : "<<r<<endl;
	cout<<"S    : "<<s<<endl;
	cout<<"Y    : "<<y<<endl;
	cout<<"G    : "<<g<<endl;

	return 0;
}
```

**client.cpp**

```
#include <iostream>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include "stdio.h"
#include <stdlib.h>
#define SA struct sockaddr
#define typeL unsigned long
using namespace std;

typeL powermod(typeL a, typeL b, typeL  q)
{
	typeL res=1;
	for(typeL i=0;i<b;i++)
		res=(res*a)%q;
	return res;
}

typeL inverse(typeL k , typeL x)
{
	typeL d=x, r=k, t, q[100],p[100];
	int i=0;
	do{
		t=d;
		q[i]=d/r;
		d=r;
		r=t%r;
		if(i==0 || i==1) p[i]=i;
		else p[i]=(p[i-2]-p[i-1]*q[i-2])%x;
		i++;
	}while(r!=0);
	p[i]=(p[i-2]-p[i-1]*q[i-2])%x;
	return p[i];
}

typeL H(typeL M)
{
	return (M^1234); //hash key=1234
}

typeL f3(typeL s,typeL q)
{
	return inverse(s,q)%q;
}

typeL f4(typeL y,typeL p,typeL q,typeL g,typeL hashval, typeL w,typeL r)
{
	typeL u1,u2;
	u1=(H(hashval)*w)%q;
	u2=(r*w)%q;
	return ( powermod(g,u1,p) * powermod(y,u2,p) )%q;
```

```
}

int main()
{
	int port;
	char addr[100]={'\0'};
	cout<<"Address  : "; scanf("%s",addr);
	cout<<"Port     : "; cin>>port;

	srand(time(NULL));
	typeL p,q,r,s,w,v,g,hashval,y;
	cout<<"p = "; cin>>p;
	cout<<"q = "; cin>>q;

	// ****Connection
	struct sockaddr_in server={AF_INET, htons(port), inet_addr(addr)};
	int sockfd = socket(AF_INET, SOCK_STREAM,0);
	connect(sockfd, (SA*)&server, sizeof(server));
	// ****Connection Established


	recv(sockfd, &hashval, sizeof(hashval), 0);
	recv(sockfd, &r, sizeof(r), 0);
	recv(sockfd, &s, sizeof(s), 0);
	recv(sockfd, &g, sizeof(g), 0);
	recv(sockfd, &y, sizeof(y), 0);

	cout<<"Packet received with values"<<endl;
	cout<<"Hash : "<<hashval<<endl;
	cout<<"R    : "<<r<<endl;
	cout<<"S    : "<<s<<endl;
	cout<<"Y    : "<<y<<endl;
	cout<<"G    : "<<g<<endl;

	//Verifying
	w=f3(s,q);
	v=f4(y,p,q,g,hashval,w,r);
	if(v==r) cout<<"Digital Signature Verified"<<endl;
	else cout<<"Digital Signature is invalid"<<endl;

	return 0;
}
```

**Output:**

**server -**
Address : 127.0.0.1
Port : 6000
p = 71
q = 7
M = 44
Packet sent with values
Hash : 1278
R : 2
S : 6
Y : 20
G : 20

**client -**
Address : 127.0.0.1
Port : 6000
p = 71
q = 7
Packet received with values
Hash : 1278
R : 2
S : 6
Y : 20
G : 20
Digital Signature Verified