NIS

LAB - 5

ROll NO. : CE146
Name : Shingala Shubham P.
Id NO. : 19CEUOS159.

* AIM : write a program to implement Knapsack cryptosystem: key generation, Encryption, Decryption.

→ source code:

```
#include <bits/stdc++.h>
using namespace std;

int gcd ( int a, int b)
{
    if ( b == 0)
        return a;
    return gcd ( b, a % b);
}

void print (vector<int> v)
{
    for (int i=0 ; i< v.size() ; i++)
        cout << v[i] << "  ";
    cout << endl;
}
```

```
int multiplicativeInverse(int a, int b)
{
    int q, r, t, t1 = 0, t2 = 1, r1 = b, r2 = a;
    while (r2 > 0)
    {
        q = r1 / r2;
        r = r1 - q * r2;
        r1 = r2;
        r2 = r;
        t = t1 - q * t2;
        t1 = t2;
        t2 = t;
    }
    if (r1 == 1)
    {
        if (t1 < 0)
            t1 += b;
        return t1;
    }
    else
        return -1;
}
```

```
bool isIncreasingSequence(vector<int> v)
{
    int sum = v[0];
    for(int i=1; i< v.size() ; i++)
    {
        if(sum >= v[i])
            return false;
        else
            sum += v[i];
    }
    return true;
}

vector<int> DecToBinary(int n)
{
    vector<int> binaryNum;
    while(n>0)
    {
        binaryNum.insert(binaryNum.begin()
                , n%2);
        n /=2;
    }
    return binaryNum;
}
```

```
vector<int> encrypt (int message, vector<int>
                                        q)
{
    vector<int> X = DecToBinary (message),
                                    sym;
    cout<<" Binary of message '" << message
        <<" ' = ";
    Print(x);
    while ( X.size() % q.size() != 0)
        X.insert( X.begin(), 0 );
    for ( int i=0, k=0 ; k< X.size(); i++)
    {
        sym.push_back (0);
        For( int j=0 ; j< q.size() &&
            k < X.size() ; j++, k++)
        {
            if (X[k])
                sym[i] += q[j];
        }
    }
    return sym;
}
```

③

```
int decrypt (int w, int m, vector<int> d_dash,
        int n, vector<int> encryption)
{
    int w_inverse = multiplicativeInverse(w,m);
    if (w_inverse != -1)
    {
        cout << " multiplicative inverse of 'w' ="
            << w_inverse << endl;
        string x = " ";
        For(int i= encryption.size()-1; i>=0;
                                    i--)
        {
            int sum = (w_inverse * encryption
                    [i]) % m;
            For(int i=n-1; i >=0; i--)
            {
                if(sum >= d_dash[i])
                {
                    x = "1" + x;
                    sum -= d_dash[i];
                }
                else
                    x = "0" + x;
            }
            if (sum != 0)
            {
                cout << " in Decryption not
                        possible in ";
                return -1;
            }
        }
```

```
        cout << " Binary of Decryption message = "
             << X << endl;
        return    stoi(X, 0, 2); // convert
                                 binary to decimal.
    }
    else {
        cout << " \n multiplicative inverse of
                 'w' is not possible ! \n";
        return  -1;
    }

}

int main()
{
    int n;
    cout << " Enter total number of item
             in 'a' :- ";
    cin >> n;
    vector <int> a_dash (n, 0);
    cout << " Enter super increasing
             sequence for 'a' :- ";
    for ( int i=0 ; i<n ; i++ )
        cin >> a_dash [i];
```

④

```
if (isIncreasingSequence (a_dash))
{
    int m = accumulate (a_dash.begin(),
            a_dash.end(), 1), w = 2;
    // m = Σ_{i=0}^{i=n} a' + 1

    while (gcd(m, w) != 1)
        w++;
    vector<int> a(n);
    for (int i=0; i<n; i++)
        a[i] = (w * a_dash[i]) % m;

    cout << "\n Private Key: \n m = " << m
    << endl << " w = " << w << "\n a_dash
    = ";
    Print (a_dash);
    cout << "\n Public Key: \n a = ";
    Print (a);

    int msg = 1;
    cout << "\n Enter message :- ";
    cin >> msg;

    cout << "\n Encryption : \n";
    vector<int> cipher = encrypt(msg,a);
    Print cout<< " Encryption of '" <<
        msg <<"' is :- ";
    Print (cipher);
```

```
cout<< "\n Decryption : \n";
cout<< ". Decryption of message is:-"
    << decrypt(w, m, a_dash, n, cipher)
    << endl;
}
else
    cout<<"\n Entered sequence is
        not increasing sequence !"
        << endl;
return 0;
}
```

→ Test case - 1 :

input: n = 4
a' = 12 32 68 143

output: private key :
m = 256
w = 3
a_dash = 12 32 68 143

Public key :
a = 36 96 204 173

Enter message :- 2342

⑤

Encryption:

Binary of message '2342' = 1 0 0 1 0 0 1 0 0 1 1 0

Encryption of '2342' is :- 209 204 300

Decryption:

Binary of decryption message = 1 0 0 1 0 0 1 0 0 1 1 0

Decryption of message is :- 2342.

→ Test-case -2 :

Input: n = 5

a' = 1 5 78 34 23

Output: Entered sequence is not Increasing sequence !

→ Test case - 3 :

Input: n = 6

a' = 23 49 133 278 532 983

Private key :

m = 1999

w = 2

a' = 23 49 133 278 532 983

Pubiic key :

a = 46 98 266 556 1064 1966

message = 9837432

Encryption :

Binary of `9837432' = 100101110000110
110111111000.

Encryption is : 2568 2012 2834 410

Decryption :

Binary of decryption message = 100101110000
110110111111000

Decrypt message : 9837432.

→ For same key if message = 145
then

Encryption :

Binary of message : 10010001

Encrypt message : 1064 2064

Decryption :

Binary of Decrypt message : 0000100100001

Decrypt message : 145