Assignment – 2

Q1.

```
#include<bits/stdc++.h>
using namespace std;
int find_MI(int x1, int x2){

    bool x = false;
        int c = 1;
        while(x != true){
            if((x1*c)%x2 == 1){
                x = true;
            }else{
                c++;
            }
        }
    return c;
}
int gcd(int a, int b)
{
    // Find Minimum of a and b
    int result = min(a, b);
    while (result > 0) {
        if (a % result == 0 && b % result == 0) {
            break;
        }
        result--;
```

```
    }

    // Return gcd of a and b
    return result;
}
bool if_coprime(vector<int> m){
    for(int i=0;i<m.size()-1;i++){
        for(int j=i+1;j<m.size();j++){
            if(gcd(m[i], m[j])!=1){
                return false;
            }
        }
    }

    return true;
}
int main()
{
    int n;
    vector<int> a;
    cout<<"Enter the number of remainders: "<<endl;
    cin>>n;
    cout<<"Enter the remainders: "<<endl;
    for(int i=0;i<n;i++){
        int temp;
        cin>>temp;
```

```
    a.push_back(temp);
}

cout<<"Enter the moduli: "<<endl;
vector<int> m;
for(int i=0;i<n;i++){
    int temp;
    cin>>temp;
    m.push_back(temp);
}

if(if_coprime(m)){
    int M=1;
    for(int i=0;i<n;i++){
        M = M * m[i];
    }

    vector<int> M_cap;
    for(int i=0;i<n;i++){
        int temp;
        temp = (M/m[i]);
        cout<<temp<<" ";
        M_cap.push_back(temp);
    }

    int X=0;
```

```
    for(int i=0;i<n;i++){
        // cout<<"M.I"<<find_MI(M_cap[i], m[i])<<" ";
        X = (a[i]*M_cap[i]*find_MI(M_cap[i], m[i])) + X;
    }

    cout<<"The unique solution is: "<<X%M<<endl;
  }else{
    cout<<"The numbers are not coprime"<<endl;
  }


}
```

Q2.

```
#include<bits/stdc++.h>
using namespace std;
int determinant(int n , vector<vector<int>> a) {
  int det = (a[0][0]*(a[1][1]*a[2][2] - a[1][2]*a[2][1])) - (a[0][1]*(a[1][0]*a[2][2] - a[2][0]*a[1][2])) + (a[0][2]*(a[1][0]*a[2][1] - a[2][0]*a[1][1]));
  return det;
}
int find_MI(int x1, int x2){

  bool x = false;
```

```
        int c = 1;
        while(x != true){
            if((x1*c)%x2 == 1){
                x = true;
            }else{
                c++;
            }
        }
    return c;
}
int main()
{
    int n;
    cout<<"Enter the size of matrix: "<<endl;
    cin>>n;

    int q;
    cout<<"Enter the mod value: "<<endl;
    cin>>q;

    cout<<"Enter the elements of the matrix: "<<endl;
    vector<vector<int>> mat;
    for(int i=0;i<n;i++){
        vector<int> temp;
        for(int j=0;j<n;j++){
            int p;
```

```
        cin>>p;
        temp.push_back(p);
    }
    mat.push_back(temp);
}

int det = determinant(n,mat);
if(det < 0){
    int x = det*-1;
    det = q - (x%q);
}

cout<<"Det: "<<det<<endl;

// Calulating the matrix of cofactors
vector<vector<int>> cof;

for(int i=0;i<n;i++){
    vector<int> temp;
    for(int j=0;j<n;j++){
        int x = i;
        int y = j;
        int x1,x2;
        int y1,y2;
        if(x == 0){
            x1 = 1;
```

```
            x2 = 2;
        }else if(x == 1){
            x1 = 0;
            x2 = 2;
        }else{
            x1 = 0;
            x2 = 1;
        }

        if(y == 0){
            y1 = 1;
            y2 = 2;
        }else if(y == 1){
            y1 = 0;
            y2 = 2;
        }else{
            y1 = 0;
            y2 = 1;
        }
        int val = mat[x1][y1]*mat[x2][y2] - mat[x1][y2]*mat[x2][y1];
        if((x+y)%2 != 0){
            val = val*-1;
        }
        temp.push_back(val);
    }
    cof.push_back(temp);
```

```
}


// Transpose matrix
vector<vector<int>> trans;

for(int i=0;i<n;i++){
    vector<int> temp;
    for(int j=0;j<n;j++){
        temp.push_back(cof[j][i]);
    }
    trans.push_back(temp);
}

for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        cout<<trans[i][j]<<" ";
    }
    cout<<endl;
}

int det_inv = find_MI(det,q);

for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        trans[i][j] = (trans[i][j] * det_inv);
```

```
            if(trans[i][j] < 0){
                int x = trans[i][j]*-1;
                trans[i][j] = q - (x%q);
            }else{
                trans[i][j] = (trans[i][j] * det_inv)%q;
            }
        }
        cout<<endl;
    }

    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            cout<<trans[i][j]<<" ";
        }
        cout<<endl;
    }

}
```

Q3.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int a,n,k;
```

```
cout<<"Enter the base value : "<<endl;
cin>>a;
cout<<"Enter the exponent value : "<<endl;
cin>>k;
cout<<"Enter the value of n : "<<endl;
cin>>n;
int c = k;
vector<int> binary;
while(c>0)
{
   int temp = c%2;
   binary.push_back(temp);
   c = c/2;

}
int size = binary.size();
for(int i=0;i<size;i++)
{
   cout<<binary[i];
}

int A,b,K;

A = a;
b = 1;
K = 0;
```

```
    for(int i=1;i<=size;i++)
    {
        K = binary[i];
        A = (A*A)%n;
        if(K == 1)
        {
            b = (A*b) % n;
        }
    }
    cout<<endl<<"Ans = "<<b;
}
```

Q4.

```
#include<bits/stdc++.h>
using namespace std;
int gcd(int a, int b)
{
    // Find Minimum of a and b
    int result = min(a, b);
    while (result > 0) {
        if (a % result == 0 && b % result == 0) {
            break;
        }
        result--;
```

```
    }

    // Return gcd of a and b
    return result;
}
int main()
{
    int n;
    cout<<"Enter the value Zn"<<endl;
    cin>>n;

    vector<int> z_star;
    for(int i=1;i<=n-1;i++){
        if(gcd(i,n) == 1){
            z_star.push_back(i);
        }
    }

    cout<<z_star.size()<<endl;
}
```

Q5.

```
#include<bits/stdc++.h>
using namespace std;
int gcd(int a, int b)
```

```
{
    // Find Minimum of a and b
    int result = min(a, b);
    while (result > 0) {
        if (a % result == 0 && b % result == 0) {
            break;
        }
        result--;
    }

    // Return gcd of a and b
    return result;
}
int main()
{
    int n;
    cout<<"Enter the value Zn"<<endl;
    cin>>n;

    vector<int> z_star;
    for(int i=1;i<=n-1;i++){
        if(gcd(i,n) == 1){
            z_star.push_back(i);
        }
    }
    // for(int i=0;i<z_star.size();i++){
```

```
//     cout<<z_star[i]<<" ";
// }

// cout<<endl;

int phi = z_star.size();
// cout<<phi<<endl;
vector<int> t;
for(int i=1;i<=phi;i++){
   if(phi%i == 0){
      // cout<<i<<" ";
      t.push_back(i);
   }
}

vector<int> order;
for(int i=0;i<z_star.size();i++){
   int q1 = z_star[i];
   for(int j=0;j<t.size();j++){
      int q2 = t[j];
      long long int res = pow(q1,q2);
      if(res%n == 1){
         cout<<"Order "<<q1<<"-"<<q2<<endl;
         order.push_back(q2);
         break;
      }
```

```
        }
    }



}
```