#include<stdio.h>

#include<math.h>

#include<stdbool.h>

#define EPSILON 0.001

int n;

int flag;

void print(int iteration,float values[])

{

  printf("Iteration %d ",iteration);

  for(int i=0;i<n;i++)

    printf("value[x%d]=%f ",i+1,values[i]);

  printf("\n");

}

int main()

{

    int i,j,k,maxIterations,iteration;

    float ratio;

    printf("Enter no of Unknowns: ");

    scanf("%d",&n);

    printf("Enter no. of iterations: ");

    scanf("%d",&maxIterations);

    float a[n][n+1];

    float values[n];;

    printf("Enter the Augmented Matrix\n");

    for(int i=0;i<n;i++)

    {

     for(int j=0;j<n+1;j++)

        scanf("%f",&a[i][j]);

    }

   printf("\n\nThe values of x,y,z are:\n");

   for(int i=0;i<n;i++)

     values[i]=0;

    float sum=0;

    float values\_new[n];

    float values\_old[n];

    for(int i=0;i<n;i++)

     values\_new[i]=0;

    for(iteration=1;iteration<=maxIterations;iteration++)

    {

        for(i=0;i<n;i++)

        {

            sum=0;

            for(j=0;j<n;j++)

            {

                if(i!=j)

                sum+=a[i][j]\*values\_new[j];

            }

            values\_new[i]=(a[i][n] - sum)/a[i][i];

        }

        //Now you have found the values of n unknowns for above iteration

        //Now check if your matching criteria satisfied , comparing with previous iteration values

        for(k=0;k<n;k++)

        {

            if(fabs(values\_old[k]-values\_new[k])<EPSILON)

                continue;

            else

            {

               flag=1;

               break;

            }

        }

        if(flag==0)

        {

            print(iteration,values\_new);  // print final values of unknowns  and return

            return 0;

        }

        flag=0; //resetting the flag

        print(iteration,values\_new); //To print intermediate roots

        //copy new values of unknowns to old value array

        for(k=0;k<n;k++)

             values\_old[k]=values\_new[k];

    } //end of iteration loop

    print(iteration,values\_new) ;

    //end of findValues()

    return 0;

}