LINEAR ALGEBRA PROJECT

Topic: Row Reduction Algorithm

**Done by:**

Prameela Nannapaneni

AP20110010120

Group – 2

CSE - B

**Row Reduction:**

Explain the row reduction algorithm and use it to find the determinant of a matrix. Also, write a program for the row reduction algorithm. The input should be an augmented matrix and the output the set of solutions. Additionally, write a program that finds the determinant of a matrix using the row reduction algorithm

**STEPS INVOLVED:**

* Input an augmented matrix.
* Perform necessary row reductions.
* Find the determinant.
* Find the rank of the matrices.
* Output the type and set of solutions.

**EXPLANATION:**

* Make all the elements in the lower triangle of the augmented matrix 0 using row operations. Similarly, make all the elements in the upper triangle 0. Now we have a diagonal matrix.
* Find the determinant by multiplying all the diagonal elements.
* Now reduce the diagonal matrix form to the identity matrix form by performing the necessary row operations.
* Find the rank of the coefficient matrix by finding the number of non-zero rows in the matrix. Similarly, find the rank of the augmented matrix also.
* If the rank of augmented matrix = rank of coefficient matrix = no of rows in the coefficient matrix, then it has a unique solution.

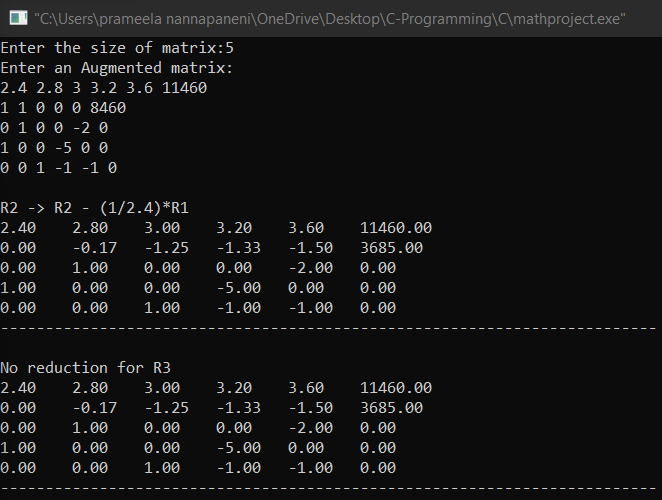
The elements in the last column of the augmented matrix are the solutions of the given equations.

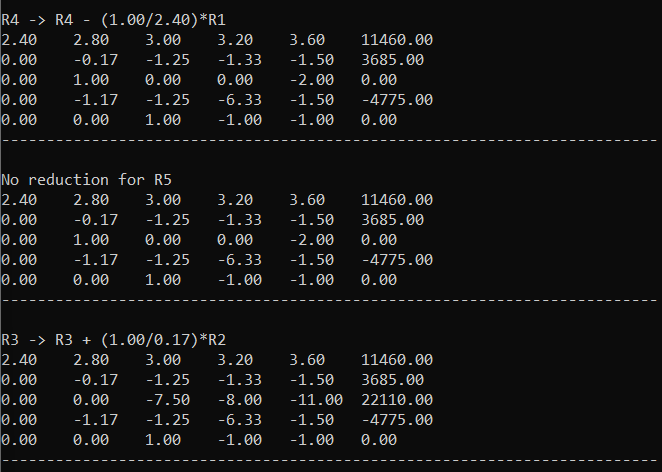
* If the rank of the augmented matrix = the rank of the coefficient matrix but not equal to no of rows in the coefficient matrix, then it has infinite solutions.
* If the rank of the augmented matrix is not equal to the rank of the coefficient matrix, then it has no solution.
* Print the determinant, type of solution, and solution set(if it is a unique solution).

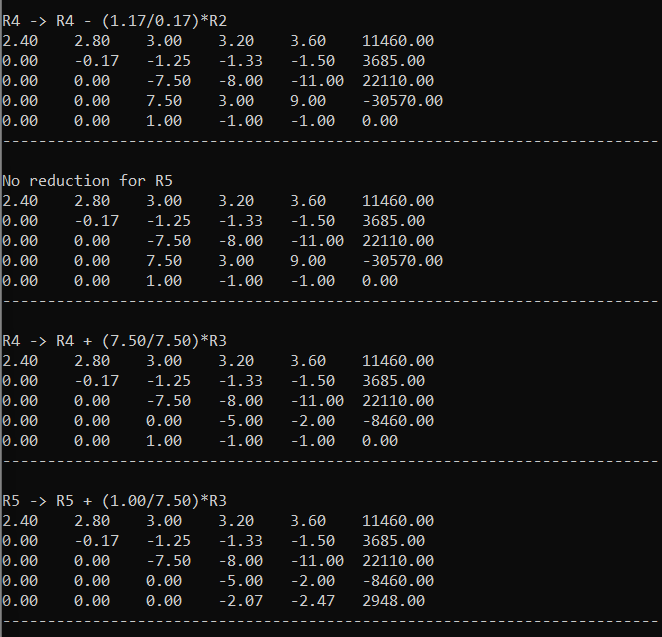
**REAL-LIFE EXAMPLES:**

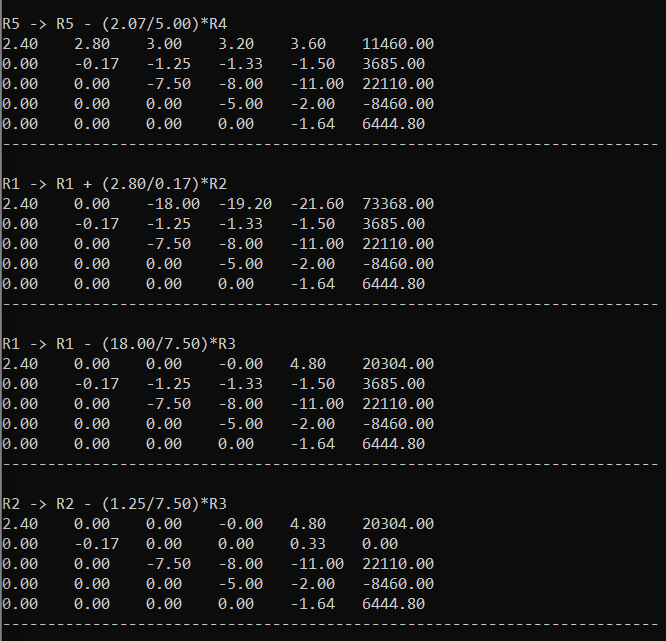
1. A gas station sells 4 types of gas: Needed for $2.40 a gallon, Regular for $2.80 a gallon, Booster for $3.00 a gallon, Performance Plus for $3.20 a gallon, and Premium for $3.60 a gallon. On a particular day, 3900 gallons of gas were sold for a total of $11460. The total amount of needed and regular sold was $8460. Two times as many gallons of Regular as Premium gas was sold. Three times as many gallons of Needed as Performance Plus gas was sold. The Sum of Premium and Performance Plus was the total Booster sold. How many gallons of each type of gas were sold that day?

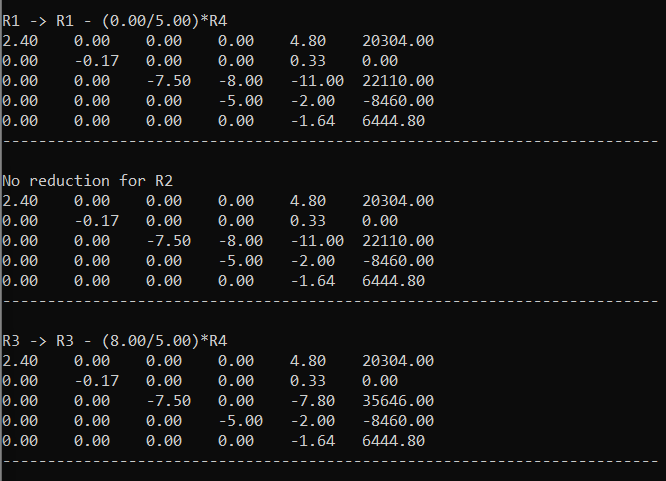
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Needed | Regular | Booster | Performance Plus | Premium | Total |
| R1 | 2.4 | 2.8 | 3 | 3.2 | 3.6 | 11460 |
| R2 | 1 | 1 | 0 | 0 | 0 | 8460 |
| R3 | 0 | 1 | 0 | 0 | -2 | 0 |
| R4 | 1 | 0 | 0 | -5 | 0 | 0 |
| R5 | 0 | 0 | 1 | -1 | -1 | 0 |

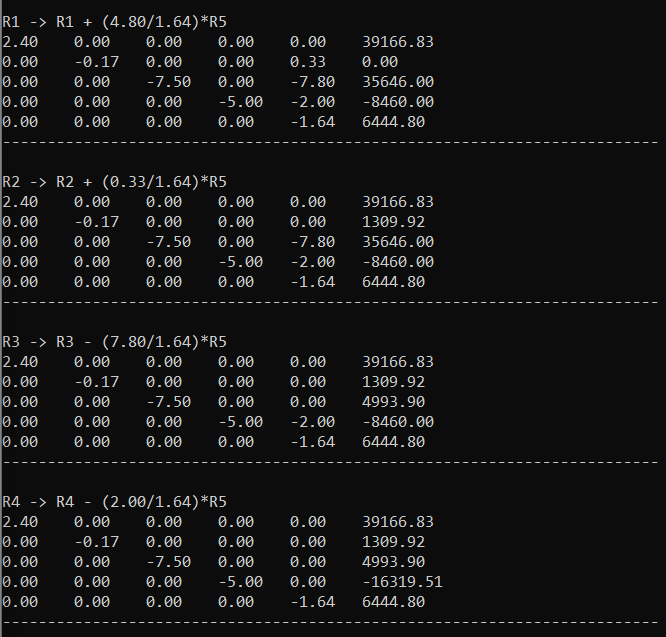


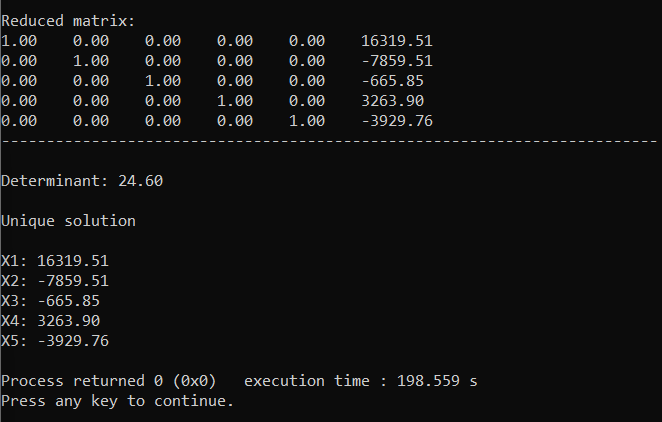






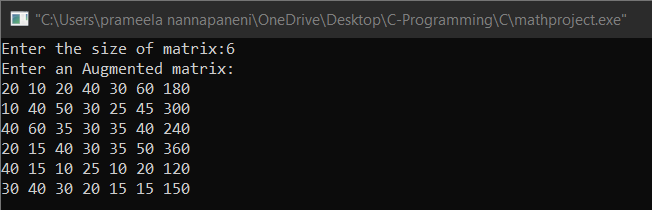


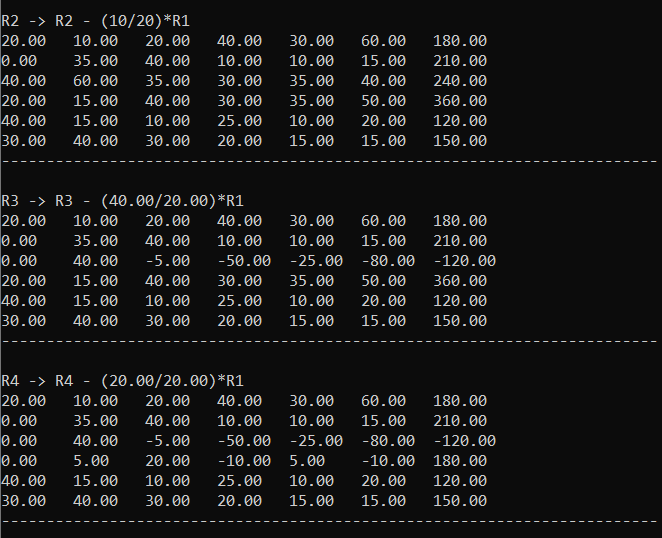


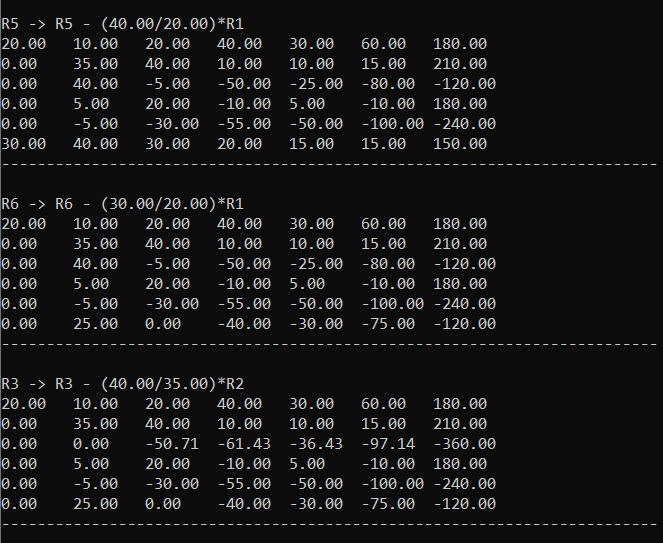


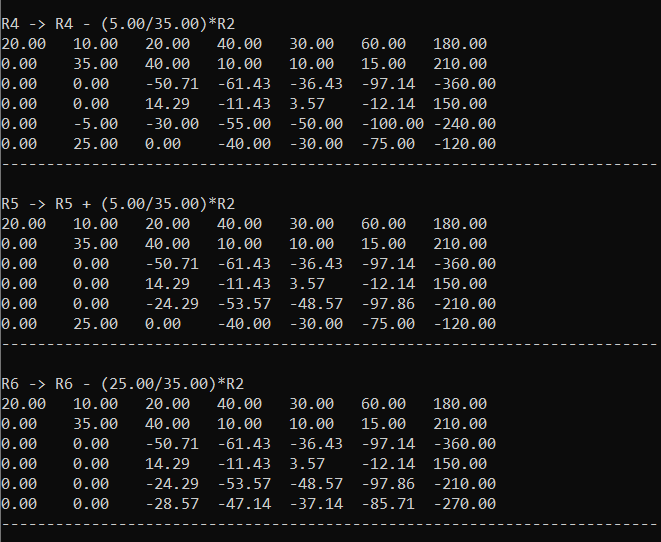
1. Chand Novelty wishes to produce 6 types of toys: types A, B, C, D, E, and F. Manufacturing a type A toy requires 20 min on machine I, 80 min on machine II, and 40 min on machine III, 70 min on machine IV, 40 min on machine V and 30 min on machine VI. Manufacturing a type B toy requires 10 min on machine I, 40 min on machine II, and 60 min on machine III, 75 min on machine IV, 15 min on machine V and 40 min on machine VI. Manufacturing a type C toy requires 20 min on machine I, 50 min on machine II, and 35 min on machine III, 80 min on machine IV, 10 min on machine V and 30 min on machine VI. Manufacturing a type D toy requires 40 min on machine I, 60 min on machine II, and 30 min on machine III, 60 min on machine IV, 25 min on machine V and 20 min on machine VI. Manufacturing a type E toy requires 30 min on machine I, 25 min on machine II, and 35 min on machine III, 35 min on machine IV, 10 min on machine V and 15 min on machine VI. Manufacturing a type F toy requires 60 min on machine I, 45 min on machine II, and 40 min on machine III, 40 min on machine IV, 20 min on machine V and 15 min on machine VI. There are 3 hours available on machine I, 5 hours available on machine II, 4 hours available on machine III, 6 hours available on machine IV, 2 hours available on machine V, and 2.5 hours available on machine VI for processing the order. How many toys of each type should Chand Novelty make in order to use all of the available time?

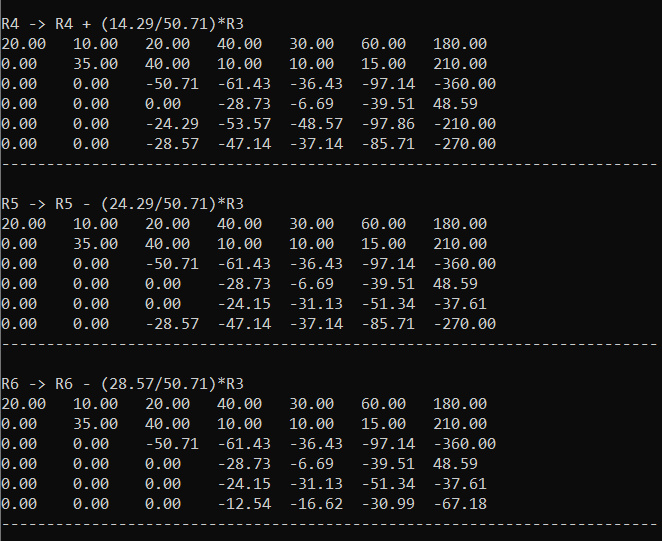
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | Total time |
| Machine 1 | 20 | 10 | 20 | 40 | 30 | 60 | 180 |
| Machine 2 | 10 | 40 | 50 | 30 | 25 | 45 | 300 |
| Machine 3 | 40 | 60 | 35 | 30 | 35 | 40 | 240 |
| Machine 4 | 20 | 15 | 40 | 30 | 35 | 50 | 360 |
| Machine 5 | 40 | 15 | 10 | 25 | 10 | 20 | 120 |
| Machine 6 | 30 | 40 | 30 | 20 | 15 | 15 | 150 |

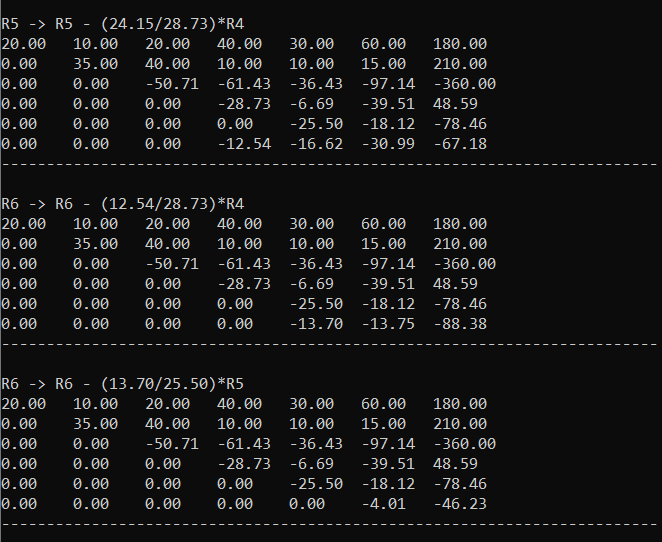


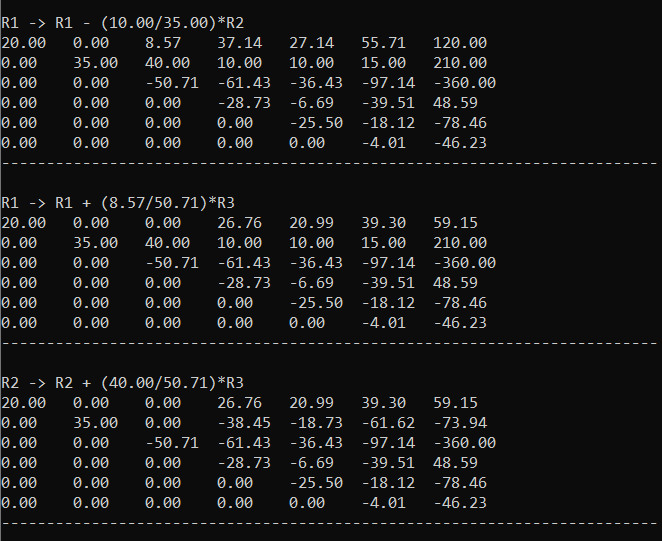


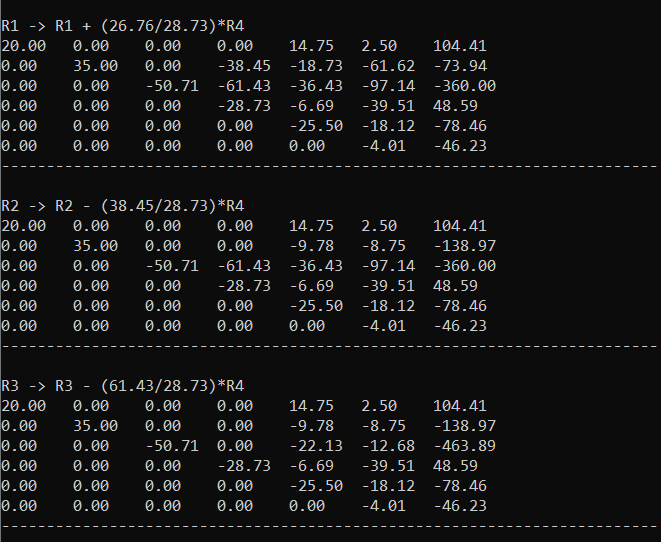


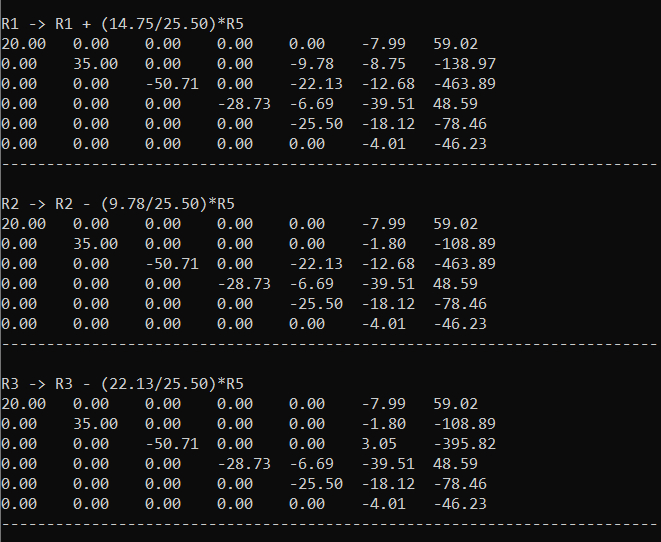


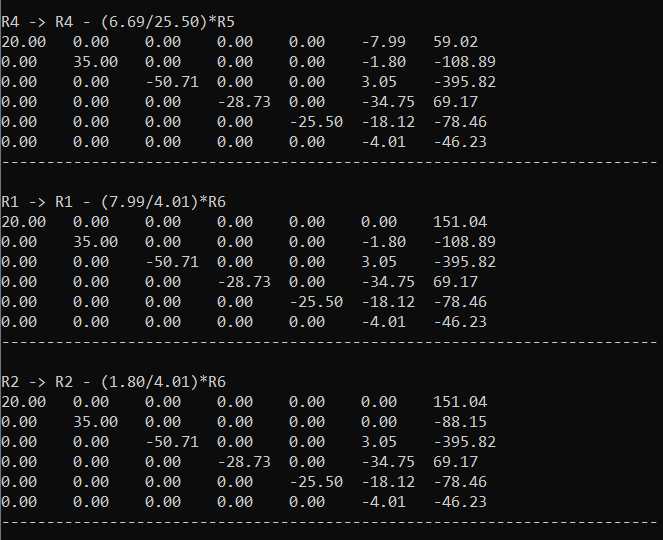


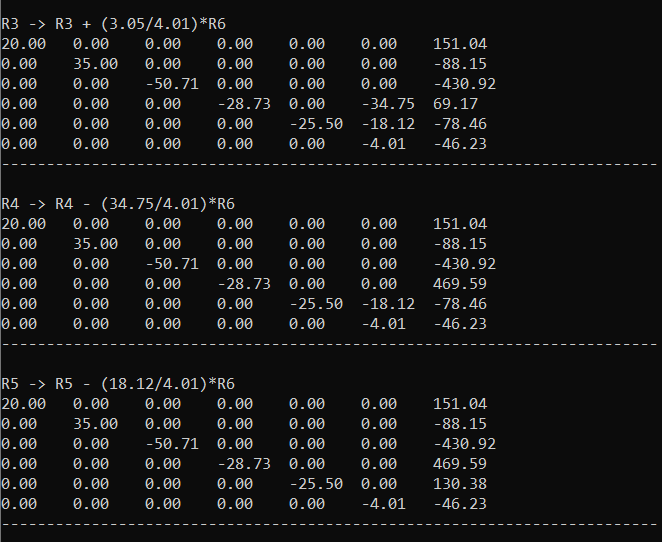


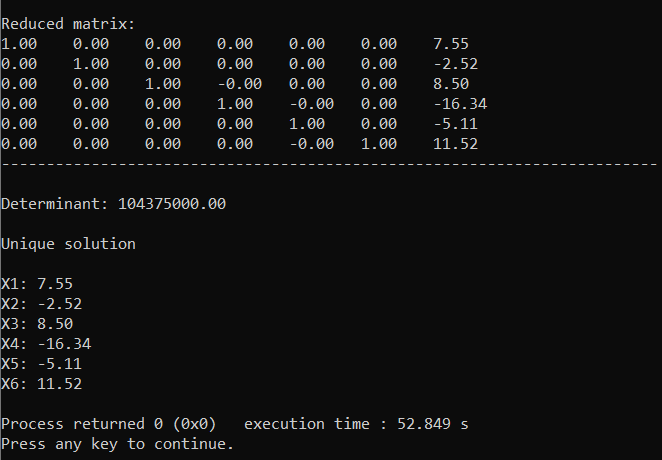












**CODE:**

#include <bits/stdc++.h>

using *namespace* std;

*int* pr=0;

*void* display(*int* *n*,*long* *double* \*\**a*,*int* *f*){

    for(*int* k=0;k<*n*;k++){

*int* q=k;

        if(!*a*[k][k]){

            for(*int* h=k+1;h<*n*;h++)

            if(*a*[h][k]){

                q=h;

                break;

            }

*double* b[*n*];

            for(*int* i=0;i<*n*+1 && q!=k;i++){

                b[i]=*a*[k][i];

*a*[k][i]=*a*[q][i];

*a*[q][i]=b[i];

            }

            pr+=1;

        }

        if(q==k && !*a*[k][k])

            continue;

        for(*int* i=0;i<*n*;i++){

            if((i<=k && !*f*)||(i>=k && *f*))

            continue;

*long* *double* p=(*a*[i][k]/*a*[k][k]);

*long* *double* l=*a*[i][k],m=*a*[k][k];

            for(*int* j=k;j<*n*+1;j++)

*a*[i][j]=*a*[i][j]-p\**a*[k][j];

            if(!p)

            cout<<endl<<"No reduction for R"<<i+1<<endl;

            else if(p>=0){

                if(l<0){

                    l\*=-1;

                    m\*=-1;

                }

                cout<<endl<<"R"<<i+1<<" -> "<<"R"<<i+1<<" - ("<<l<<"/"<<m<<")\*R"<<k+1<<endl;

            }

            else{

                if(l<0)

                l\*=-1;

                else

                m\*=-1;

                cout<<endl<<"R"<<i+1<<" -> "<<"R"<<i+1<<" + ("<<l<<"/"<<m<<")\*R"<<k+1<<endl;

            }

            for(*int* i=0;i<*n*;i++){

                for(*int* j=0;j<*n*+1;j++)

                cout<< fixed <<setprecision(2) <<*a*[i][j]<<"\t";

                cout<<endl;

            }

            cout<<"-------------------------------------------------------------------------\n";

        }

}

}

*int* main(){

*long* *int* n,f=0;

    cout<<"Enter the size of matrix:";

    cin>>n;

*long* *double* \*\*a,z,mul=1;

    a = new *long* *double* \*[n];

    for(*int* i = 0; i <n; i++)

    a[i] = new *long* *double*[n+1];

    cout<<"Enter an Augmented matrix:\n";

    for(*int* i=0;i<n;i++)

        for(*int* j=0;j<n+1;j++)

        cin>>a[i][j];

    display(n,a,0);

    display(n,a,1);

    for(*int* i=0;i<n;i++)

        mul\*=a[i][i];

    for(*int* i=0;i<n;i++)

        if(a[i][i]){

            for(*int* j=0;j<n+1;j++)

                if(i!=j && a[i][j])

                a[i][j]/=a[i][i];

            a[i][i]=1;

        }

    cout<<endl<<"Reduced matrix:\n";

    for(*int* i=0;i<n;i++){

        for(*int* j=0;j<n+1;j++)

        cout<<a[i][j]<<setprecision(2)<<"\t";

        cout<<endl;

    }

    cout<<"-------------------------------------------------------------------------\n";

    mul\*=pow(-1,pr);

    cout<<endl<<"Determinant: "<<mul<<endl;

*int* c=0,d=0;

    for(*int* i=0;i<n;i++)

        for(*int* j=0;j<n;j++)

            if(a[i][j]){

                c++;

                break;

}

    for(*int* i=0;i<n;i++)

        for(*int* j=0;j<n+1;j++)

            if(a[i][j]){

                d++;

                break;

         }

    if(c==d && c==n && mul){

        cout<<endl<<"Unique solution\n";

        for (*int* i=0;i<n;i++)

        cout<<"\nX"<<i+1<<": "<<a[i][n]<<"\t";}

    else if(c==d && c<n)

    cout<<endl<<"Infinite solution\n";

    else

    cout<<endl<<"No Solution\n";

    cout<<endl;

    return 0;

}