

1. Будем говорить, что столбец  $i_2$  больше столбца  $i_1$ , если сумма разностей  $a[j][i_2] - a[j][i_1]$ ,  $j=0, \dots, N-1$ , положительна.
2. Будем говорить, что строка  $i_2$  больше строки  $i_1$ , если сумма разностей  $a[i_2][j] - a[i_1][j]$ ,  $j=0, \dots, N-1$ , положительна.
3. Назовём весом столбца матрицы число, равное среднему гармоническому абсолютных значений элементов этого столбца. Если в столбце присутствует 0 ( $<1.e-15$ ), то среднее гармоническое полагаем равным нулю.
4. Назовём весом строки матрицы число, равное среднему гармоническому абсолютных значений элементов этой строки. Если в строке присутствует 0 ( $<1.e-15$ ), то среднее гармоническое полагаем равным нулю.
5. Назовём нормой столбца матрицы число, равное сумме модулей элементов этого столбца.
6. Назовём нормой строки матрицы число, равное сумме модулей элементов этой строки.
7. Назовём нормой столбца матрицы число, равное сумме квадратов элементов этого столбца.
8. Назовём нормой строки матрицы число, равное сумме квадратов элементов этой строки.
9. Назовём весом столбца матрицы число, равное максимальному элементу этого столбца.
10. Назовём весом строки матрицы число, равное максимальному элементу этой строки.
11. Назовём весом строки матрицы число, равное разности между максимальным и минимальным элементами этой строки.
12. Назовём весом столбца матрицы число, равное разности между максимальным и минимальным элементами этого столбца.
13. Назовём весом столбца матрицы число, равное полусумме двух наибольших элементов этого столбца(наибольшие элементы могут быть равны).
14. Назовём весом строки матрицы число, равное полусумме двух наибольших элементов этих строк(наибольшие элементы могут быть равны).
15. Назовём весом столбца матрицы число, равное сумме только положительных разностей  $a[j+1][i] - a[j][i]$  соседних элементов этого столбца (т.е. возрастающих переходов от  $j$  к  $j+1$  для  $j=0, \dots, N-2$ ). Если таких разностей нет, то вес столбца матрицы положим равным нулю.
16. Назовём весом строки матрицы число, равное сумме только положительных разностей  $a[i][j+1] - a[i][j]$  соседних элементов этой строки(т.е. возрастающих переходов от  $j$  к  $j+1$  для  $j=0, \dots, N-2$ ). Если таких разностей нет, то вес строки матрицы положим равным нулю.
17. (строки) Лексикографический порядок.
18. (столбцы) Лексикографический порядок.

**19.** Назовем весом столбца матрицы число, равное максимум модулей разности двух соседних элементов этого столбца т.е  $|a[(i-1)*m+j] - a[i*m+j]|$

**20.** Назовем весом строки матрицы число, равное максимум модулей разности двух соседних элементов этой строки т.е  $|a[(i-1)*m+j] - a[i*m+j]|$

```
int main (void)
{
    int m,n,i,j;
    double*a,*c;
    FILE*in=fopen("input.txt","r");
    FILE*out=fopen("output.txt","w");
    if(in==NULL)
        return -1;
    if(out==NULL)
    {
        fclose(in);
        return -1;
    }
    if(fscanf(in,"%d",&n)!=1)
    {
        fclose(in);fclose(out);
        return -1;
    }
    if(fscanf(in,"%d",&m)!=1)
    {
        fclose(in);fclose(out);
        return -1;
    }
    if((n<0)|| (m<0))
    {
        fclose(in);fclose(out);
        return -1;
    }
    if (!(a=(double*)malloc(n*m*sizeof(double))))
    {
        fclose(in);fclose(out);
        return -1;
    }
    if (!(c=(double*)malloc(n*sizeof(double))))
    {
        free(a);fclose(in);fclose(out);
        return -1;
    }
    for(i=0;i<n;i++)
        c[i]=0;
    for(i=0;i<n*m;i++)
        if (fscanf(in,"%lf",&a[i])!=1)
        {
            free(a);free(c);fclose(in);fclose(out);
            return -1;
        }
    fclose(in);
    process(a,n,m);
    multipl(a,c,n,m);
    fprintf(out,"%d %d\n",n,m);
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            fprintf(out,"%lf ",a[i*m+j]);
        fprintf(out,"\n");
    }
    fprintf(out,"\n");
    for(i=0;i<n;i++)
        fprintf(out,"%lf\n",c[i]);
    free(a);
    free(c);
    fclose(out);
    return 0;
}
```

## Общая

```
void multipl(double*a,double*c,int n,int m)
```

```
{
    int i,j;
    double g=0;
    for (i=0;i<n;i++)
    {
        g=0;
        for(j=0;j<m;j++)
            g+=a[i*m+j]*a[m*(n-1)+j]; //a[j] если первая строка
        c[i]=g;
    }
}
```

## Для столбцов

```
void process(double*a,int n,int m)
```

```
{
    int i,k,j;
    double w;
    for(k=0;k<m-1;k++)
    {
        for(j=0;j<m-1;j++)
        {
            if(weight(a,n,m,j)>weight(a,n,m,j+1))
                for(i=0;i<n;i++)
                {
                    w=a[i*m+j];
                    a[i*m+j]=a[i*m+(j+1)];
                    a[i*m+(j+1)]=w;
                }
        }
    }
}
```

## Для строк

```
void process(double*a,int n,int m)
```

```
{
    int i,k,j;
    double w;
    for(k=0;k<n-1;k++)
    {
        for(i=0;i<n-1;i++)
        {
            if(weight(a,m,i)>weight(a,m,i+1))
                for(j=0;j<m;j++)
                {
                    w=a[i*m+j];
                    a[i*m+j]=a[(i+1)*m+j];
                    a[(i+1)*m+j]=w;
                }
        }
    }
}
```

---

### 1.

```
double weight(double*a,int n,int m,int j)
```

```
{
    int i;
    double sum=0;
    for(i=0;i<n;i++)
        sum+=a[i*m+j];
    return sum;
}
```

### 2.

```
double weight(double*a,int m,int i)
```

```
{
    int j;
    double sum=0;
}
```

```

        for(j=0;j<m;j++)
            sum+=a[i*m+j];
    return sum;
}
3.
#define e 1.e-15
double weight(double*a,int n,int m,int j)
{
    int i;
    double sum=0;
    double garm=0;
    for(i=0;i<n;i++)
    {
        if ((fabs(a[i*m+j])>e))
            sum+=(1/fabs(a[i*m+j]));
        else
        {
            sum=0;
            break;
        }
    }
    if (fabs(sum)>e)
        garm=n/sum;
    return garm;
}

```

```

4.
#define e 1.e-15
double weight(double*a,int m,int i)
{
    int j;
    double sum=0;
    double garm=0;
    for(j=0;j<m;j++)
    {
        if (a[i*m+j]!=0)
            sum+=(1/fabs(a[i*m+j]));
        else
        {
            sum=0;
            break;
        }
    }
    if (fabs(sum)>e)
        garm=m/sum;
    return garm;
}

```

## 5. столбцы

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double sum=0;
    for(i=0;i<n;i++)
        sum+=fabs(a[i*m+j]);
    return sum;
}

```

## 6. строки

```

double weight(double*a,int m,int i)
{
    int j;
    double sum=0;
    for(j=0;j<m;j++)
        sum+=fabs(a[i*m+j]);
    return sum;
}

```

```

7.
double weight(double*a,int m,int i)

```

```

{
    int j;
    double s=0;
    for(j=0;j<m;j++)
    {
        s+=a[i*m+j]*a[i*m+j];
    }
    return s;
}

```

8.

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double s=0;
    for(i=0;i<n;i++)
    {
        s+=a[i*m+j]*a[i*m+j];
    }
    return s;
}

```

9. не уверена!

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double max=a[j];
    for(i=0;i<n;i++)
    {
        if (a[i*m+j]>max)
            max=a[i*m+j];
    }
    return max;
}

```

10.

```

double weight(double*a,int m,int i)
{
    int j;
    double max=a[i*m];
    for(j=0;j<m;j++)
    {
        if (a[i*m+j]>max)
            max=a[i*m+j];
    }
    return max;
}

```

11.строки

```

double weight(double*a,int m,int i)
{
    int j;
    double max=a[i*m];
    double min=a[i*m];
    for(j=0;j<m;j++)
    {
        if (a[i*m+j]>max)
            max=a[i*m+j];
        if (a[i*m+j]<min)
            min=a[i*m+j];
    }
    return max-min;
}

```

12.

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double max=a[j];
    double min=a[j];
    for(i=0;i<n;i++)
    {

```

```

        if (a[i*m+j]>max)
            max=a[i*m+j];
        if (a[i*m+j]<min)
            min=a[i*m+j];
    }
    return max-min;
}

```

### 13. !!!!не знаю как работает (столбцы)

```

#define e 1.e-15
double weight(double*a,int n,int m,int j)
{
    int i,t,max1,max2;
    if (a[j]>a[j+m])
    {
        max1=0;max2=1;
    }
    else
    {
        max1=1;max2=0;
    }
    for (i=2;i<n;i++)
    {
        if (a[i*n+j]>a[max1*n+j])
        {
            t=max1;
            max1=i;
            if (a[t]>a[max2*n+j])
                max2=t;
        }
        else
            if (a[i*n+j]>a[max2*n+j])
                max2=i;
    }
    return ((a[max1*n+j]+a[n*max2+j])/2);
}

```

### 14. !аналогично(см.выше)

### 15.

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double sum=0,f=0;
    for(i=0;i<n-1;i++)
    {
        f=a[(i+1)*m+j]-a[i*m+j];
        if (f>0)
            sum+=f;
    }
    return sum;
}

```

### 16. строки

```

double weight(double*a,int m,int i)
{
    int j;
    double sum=0,f=0;
    for(j=0;j<m-1;j++)
    {
        f=a[i*m+(j+1)]-a[i*m+j];
        if (f>0)
            sum+=f;
    }
    return sum;
}

```

### 17. !меняется процесс

```

void process (double*a,int n,int m)
{
    int i,k,j;
    double w;
}

```

```

for (k=0;k<n-1;k++)
{
    for (i=0;i<n-1;i++)
    {
        if (weight(a,m,i)==1)
            for (j=0;j<m;j++)
            {
                w=a[i*m+j];
                a[i*m+j]=a[(i+1)*m+j];
                a[(i+1)*m+j]=w;
            }
    }
}
}

double weight (double*a,int m,int i)
{
    int j;
    for (j=0;j<m;j++)
    {
        if (a[i*m+j]>a[(i+1)*m+j])
            return 1;
        if (a[i*m+j]<a[(i+1)*m+j])
            break;
    }
    return 0;
}

```

## 18. !меняется процесс

аналогично(см.выше)

## 19.

```

double weight(double*a,int n,int m,int j)
{
    int i;
    double max=0;
    for(i=0;i<n-1;i++)
    {
        if (fabs(a[(i+1)*m+j]-a[i*m+j])>max)
            max=fabs(a[(i+1)*m+j]-a[i*m+j]);
    }
    return max;
}

```

## 20.

```

double weight(double*a,int m,int i)
{
    int j;
    double max=0;
    for(j=0;j<m-1;j++)
    {
        if (fabs(a[i*m+(j+1)]-a[i*m+j])>max)
            max=fabs(a[i*m+(j+1)]-a[i*m+j]);
    }
    return max;
}

```