

1. Maximum

Given a two-dimensional array $n \times n$. Find the largest number in array.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$) — array size. Remaining lines contain n rows with n integers $a[i][j]$ ($-10^9 \leq a[i][j] \leq 10^9$) — elements of array.

Output

One number, maximal number in array.

Solution

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n, max = -1e9 - 17;
    cin >> n;
    int arr[n][n];
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            cin >> arr[i][j];

    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            if(arr[i][j] > max)
                max = arr[i][j];
    cout << max;
    return 0;
}
```

Input	Output
5 1 8 37 3 2 9 3 84 4 89 12 4 56 3 2 43 6 23 9 2 62 98 21 8 5	98
2 12 12 12 12	12
5 12 34 90 1 23 2 37 48 36 41	94

45 89 23 84 94 65 48 35 49 56 43 75 38 47 64	
4 2 7 3 1 5 3 8 9 4 4 9 6 9 4 7 2	9
3 7 3 4 1 8 3 2 3 5	8

2. The second

Given a two-dimensional array $n \times n$. Find the second largest number in array. If all elements are equal, output 0.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$) — array size. Remaining lines contain n rows with n integers $a[i][j]$ ($-10^9 \leq a[i][j] \leq 10^9$) — elements of array.

Output

One number, second maximal number in array.

Solution

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int n, max = -1e9 - 17, second = -1e9 - 17;
    cin >> n;
    int arr[n][n];
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            cin >> arr[i][j];

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(arr[i][j] > max) {
                second = max;
                max = arr[i][j];
            }
            else if(arr[i][j] > second && arr[i][j] != max)
```

```

        second = arr[i][j];
    }
}
if(second == -1e9 - 17) cout << 0;
else cout << second;
return 0;
}

```

Input	Output
4 -1 7 3 2 9 2 4 1 8 2 -6 4 1 4 9 4	8
3 1 1 1 1 1 1 1 1 1	0
5 12 34 90 1 23 2 37 48 36 41 45 89 23 84 94 65 48 35 49 56 43 75 38 47 64	90
4 2 7 3 1 5 3 8 9 4 4 9 6 9 4 7 2	8
3 7 3 4 1 8 3 2 3 5	7

3. Negatives

Given a two-dimensional array of size $n \times m$ (n rows, m columns). Write a program, which outputs count of negative numbers in array.

Input

The first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 10^5$) — array size. Remaining lines contain n rows with m integers $a[i][j]$ ($-10^5 \leq a[i][j] \leq 10^5$) — elements of array.

Output

Single integer, negative numbers count.

Solution

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, m, cnt = 0;
```

```
    cin >> n >> m;
```

```
    int arr[n][m];
```

```
    for(int i = 0; i < n; i++)
```

```
        for(int j = 0 ; j < m; j++)
```

```
            cin >> arr[i][j];
```

```
    for(int i = 0; i < n; i++)
```

```
        for(int j = 0 ; j < m; j++)
```

```
            if(arr[i][j] < 0) cnt++;
```

```
    cout << cnt;
```

```
    return 0;
```

```
}
```

Input	Output
3 4 -1 0 2 3 5 8 20 -12 7 8 -4 9	3
3 2 -3 5 0 8 -8 12	2
4 3 1 2 3 4 3 9 9 3 5 23 43 3	0
2 4 1 -3 2 8 46 37 3 9	1
2 2 0 1 -2 -5	2

4. Multiplication table

Write a program that prints an $n \times n$ multiplication table.

Input

The first line contains one integer n ($2 < n \leq 100$) — array size.

Output

Two-dimensional array. Separate numbers with one space.

Solution:

```
#include <iostream>

using namespace std;

int main()
{
    int n, num = 0;
    cin >> n;
    int arr[n][n];
    for(int i = 0; i < n; i++)
        arr[0][i] = arr[i][0] = num++;
    for(int i = 1; i < n; i++)
        for(int j = 1; j < n; j++)
            arr[i][j] = arr[0][j]*arr[i][0];

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

Input	Output
3	0 1 2 1 1 2 2 2 4
4	0 1 2 3 1 1 2 3 2 2 4 6 3 3 6 9
2	0 1 1 1
5	0 1 2 3 4

	1 1 2 3 4 2 2 4 6 8 3 3 6 9 12 4 4 8 12 16
--	---

5. Star triangle

Write a program that generates triangle made of [*].

Input

The first line contains one integer n ($1 \leq n \leq 100$) — number of rows.

Output

Triangle with n rows.

Solution

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int width;

    cin >> width;

    for (int i = 1; i <= width; i++) {
        for (int j = 1; j <= i; j++)
            cout << "[*]";
        cout << "\n";
    }
    return 0;
}
```

Input	Output
5	[*] [*][*] [*][*][*] [*][*][*][*] [*][*][*][*][*]
2	[*] [*][*]
1	[*]
3	[*] [*][*] [*][*][*]

6. Position of maximum

Given a two-dimensional array $n \times n$. Write a program, which finds position of maximum element in array. If maximums are two or more you should output position of the first one.

Input

The first line contains two integers n ($1 \leq n \leq 10^9$) and m ($1 \leq m \leq 10^9$) — array size. Remaining lines contain n rows with m integers ($-10^9 \leq a[i][j] \leq 10^9$) — elements of array.

Output

Two integers - row and column indexes (start with 1), position of maximum in array.

Solution

```
#include <iostream>
using namespace std;

int main()
{
    int n, row, col, max = -1e9 - 17;
    cin >> n;
    int arr[n][n];
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            cin >> arr[i][j];

    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            if(arr[i][j] > max) {
                max = arr[i][j];
                row = i;
                col = j;
            }
    cout << ++row << " " << ++col;
    return 0;
}
```

Input	Output
3 0 -1 2 4 8 4 -3 8 0	2 2
4 0 13 4 2 -23 8 2 0 4 85 0 2	3 2

5 48 5 2	
5 1 28 0 -3 54 43 8 5 3 9 5 4 2 0 -23 38 50 3 43 8 -4 58 3 9 2	5 2
2 8 3 0 2	1 1
3 -2 3 8 3 4 9 5 3 0	2 3

7. Dots or number

Write program that prints numbers in increasing order in diagonal and "." on other shells.

Input

The first line contains one integer n ($2 \leq n \leq 100$) — number of rows.

Output

Print numbers on diagonal.

Solution

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++)
            cout << ".";
        cout << i;
        for (int k = 1; k <= i - 1; k++)
            cout << ".";
        cout << endl;
    }
    return 0;
}
```

Input	Output
4	...1 ..2.

	.3.. 4...
51 ...2.. ..3.. .4... 5....
3	..1 .2.. 3..
1012..3..4...5....6..... ...7..... ..8..... .9..... 10.....
2	.1 2..

8. To drop or not to drop?

Help Aisultan choose which discipline to drop, so he will have a good GPA. Each row represents one subject with its grades. Pick one subject with the minimal total grades. If two or more subjects' grades are equal, pick the first one.

Input

The first line contains two integers n ($2 \leq n \leq 10$) and m ($1 \leq m \leq 10$) — number of subjects and grades respectively. Remaining lines contain n rows with m integers $a[i][j]$ ($0 \leq a[i][j] \leq 100$) — elements of array.

Output

Single integer, index of row.

Solution

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n, m, drop = 10000, cnt = 0, subj;
    cin >> n >> m;
    int arr[n][m];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> arr[i][j];
```

```

for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++)
        cnt += arr[i][j];
    if(cnt < drop) {
        drop = cnt;
        subj = i;
    }
    cnt = 0;
}
cout << ++subj;
return 0;
}

```

Input	Output
3 4 0 78 2 3 9 7 32 6 67 3 29 2	2
3 3 12 9 8 0 1 2 4 3 8	2
2 3 7 4 9 0 1 3	2
2 2 1 1 1 1	1
4 2 0 8 98 1 4 32 7 53	1

9. Don't wanna be perfect

If an element is a perfect square, replace it with its root.

Input

The first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 10^5$) — array size. Remaining lines contain n rows with m integers $a[i][j]$ ($1 \leq a[i][j] \leq 10^9$) — elements of array.

Output

Array with some replaced elements.

Solution

```
#include <iostream>
```

```

#include <cmath>

using namespace std;

int main()
{
    int n, m, y;
    cin >> n >> m;
    int arr[n][m];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> arr[i][j];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++) {
            y = sqrt(arr[i][j]);
            if (y * y == arr[i][j]) arr[i][j] = y;
        }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }
    return 0;
}

```

Input	Output
3 3 16 7 8 9 5 6 3 4 8	4 7 8 3 5 6 3 2 8
4 3 73 2 92 121 76 3 75 392 12 9 33 42	73 2 92 11 76 3 75 392 12 3 33 42
2 3 1 12 6 8 5 2	1 12 6 8 5 2
4 5 1 2 8 4 9 14 9 2 25 7 3 1 7 9 2 16 82 7 33 1	1 2 8 2 3 14 3 2 5 7 3 1 7 3 2 4 82 7 33 1
2 2	3 3

9 9 4 4	2 2
------------	-----

10. Odd or even?

Given a two-dimensional array of size $n \times m$ (n rows, m columns). Add 1 to those shells where $i + j$ is even, and subtract 1, if $i + j$ is odd. Array indexing starts from 0 (0 is even).

Input

The first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 10^5$) — array size. Remaining lines contain n rows with m integers $a[i][j]$ ($1 \leq a[i][j] \leq 10^9$) — elements of array.

Output

Two-dimensional array.

Solution

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int n, m;
    cin >> n >> m;
    int arr[n][m];
    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++)
            cin >> arr[i][j];

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            if((i + j) % 2 == 0) arr[i][j]++;
            else arr[i][j]--;
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }
    return 0;
}
```

Input	Output
3 3 8 2 4	9 1 5 11 4 3

12 3 4 7 4 9	8 3 10
3 4 16 2 9 3 3 47 5 1 2 38 4 9	17 1 10 2 2 48 4 2 3 37 5 8
2 2 32 9 2 3	33 8 1 4
2 3 8 7 2 35 47 3	9 6 3 34 48 2
4 4 1 2 49 54 34 38 53 5 5 8 63 2 37 39 75 3	2 1 50 53 33 39 52 6 6 7 64 1 36 40 74 4

11.Sum of elements

Write a program that enters a two-dimensional array and calculates the sum of its elements in columns and rows.

Input

The first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 10^5$) — array size. Remaining lines contain n rows with m integers — elements of matrix.

Output

Number of each column,row and sum of its elements.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int n, m,i, j;
    cin >> n >> m;
    int arr[n][m];
    int r[n],c[m];
    for(i = 0; i < n; i++) {
        for(j = 0 ; j < m; j++){
            cin >> arr[i][j];
        }
    }
    for ( i=0; i<n; i++)
    {
        r[i]=0;
```

```

        for ( j=0; j<m; j++)
            r[i]+=arr[i][j];
    }

    for ( j=0; j<m; j++)
    {
        c[j]=0;
        for ( i=0; i<n; i++)
            c[j]+=arr[i][j];
    }
    for ( i=0; i<n; i++)
        cout<<"\nThe sum of row number "<<i+1<<" is "<<r[i];
    for ( j=0; j<m; j++)
        cout<<"\nThe sum of column number "<<j+1<<" is "<<c[j];

    return 0;
}

```

Input	Output
3 4 1 2 4 6 2 5 7 3 1 4 5 94	The sum of row number 1 is 13 The sum of row number 2 is 17 The sum of row number 3 is 104 The sum of column number 1 is 4 The sum of column number 2 is 11 The sum of column number 3 is 16 The sum of column number 4 is 103
2 2 1 2 0 2	The sum of row number 1 is 3 The sum of row number 2 is 2 The sum of column number 1 is 1 The sum of column number 2 is 4
2 3 1 4 -1 2 5 -5	The sum of row number 1 is 4 The sum of row number 2 is 2 The sum of column number 1 is 3 The sum of column number 2 is 9 The sum of column number 3 is -6

12. Minimum numbers

Find the sum of the smallest elements of each column of the matrix and their coordinates. If there same min elements in column program takes coordinates of the first one.

Input

The first line contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 10^5$) — array size. Remaining lines contain n rows with m integers — elements of matrix.

Output

Coordinates of each smallest element in each column and their sum.

Solution

```
#include <iostream>
```

```
using namespace std;
int main()
{
    int n, m;
    cin >> n >> m;
    int a[n][m], summa=0, min, x, y;
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<m; j++)
        {
            cin >> a[i][j];
        }
    }
    cout << "coordinates of min elements: \n";
    for (int j=0; j<m; j++)
    {
        min=a[0][j];
        x=1;
        y=j+1;
        for (int i=0; i<n; i++)
        {
            if (a[i][j]<min)
            {
                min=a[i][j];
                x=i+1;
                y=j+1;
            }
        }
        summa+=min;
        cout << x << ";" << y << endl;
    }
}
```

```

    cout << "\nTheir sum:\n";
    cout << summa;
    return 0;
}

```

Input	Output
2 3 -1 2 3 -2 -4 2	coordinates of min elements: 2;1 2;2 2;3 Their sum: -4
3 4 1 5 7 4 2 3 4 5 1 2 1 0	coordinates of min elements: 1;1 3;2 3;3 3;4 Their sum: 4
4 1 1 3 2 5	coordinates of min elements: 1;1 Their sum: 1
3 4 1 2 3 4 1 3 4 6 3 2 5 7	coordinates of min elements: 1;1 1;2 1;3 1;4 Their sum: 10

13.Spiral

Return all elements of the matrix in spiral order, given a matrix of $n \times n$ elements. Numbers start increasing around the corner matrix($a[0][0]$) then goes inside up to the middle element making spiral shape.

Input

The first line contains integer n ($1 \leq n \leq 10^5$). Remaining lines contain integers $a[i][j]$ ($1 \leq a[i][j] \leq 10^9$) — elements of matrix.

Output

Spiral matrix.

Solution

```
#include<iostream>
using namespace std;

int main()
{
    int n;
    cin>>n;
    int arr[n][n];
    int len=n,k=1,p=0,i;
    while(k<=n*n)
    {
        for(i=p;i<len;i++)
        {
            arr[p][i]=k++;
        }
        for(i=p+1;i<len;i++)
        {
            arr[i][len-1]=k++;
        }
        for(i=len-2;i>=p;i--)
        {
            arr[len-1][i]=k++;
        }
        for(i=len-2;i>p;i--)
        {
            arr[i][p]=k++;
        }
        p++,len=len-1;
    }
    if(!n%2)
    {
        arr[(n+1)/2][(n+1)/2]=n*n;
    }
}
```

```

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

```

Input	Output
3	1 2 3 8 9 4 7 6 5
4	1 2 3 4 12 13 14 5 11 16 15 6 10 9 8 7

14. Prime numbers

A *prime number* (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers.

Find all prime numbers from 2 to n using a nested for loop.

Input

Integer n ($2 < n \leq 10^5$).

Output

All prime numbers

Solution

```

#include <iostream>
using namespace std;

```

```

int main () {
    int i, j;
    int n;
    cin>>n;

    for(i = 2; i<n; i++) {
        for(j = 2; j <= (i/j); j++)

```

```

        if(!(i%j)) break;
        if(j > (i/j)) cout << i << " is prime\n";
    }

    return 0;
}

```

Input	Output
50	2 is prime 3 is prime 5 is prime 7 is prime 11 is prime 13 is prime 17 is prime 19 is prime 23 is prime 29 is prime 31 is prime 37 is prime 41 is prime 43 is prime 47 is prime
10	2 is prime 3 is prime 5 is prime 7 is prime

15. Main diagonal

The *main diagonal of a matrix* consists of those elements that lie on the diagonal that runs from top left to bottom right.

Return the maximum value of the element on this diagonal and its coordinates. If there same maximum elements it takes coordinates of the first one.

Input

The first line contains integer n ($1 \leq n \leq 10^5$) (size of matrix $n \times n$). Remaining lines contain integers — elements of matrix.

Output

Integer(maximum value) and its coordinates.

Solution

```
#include<iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;
    int arr[n][n];
    int i,j;
    for(i=0;i<n;i++)
    {
        for (j=0; j<n; j++)
        {
            cin>>arr[i][j];
        }
    }
    int max = arr[0][0];
    int imax = 0;
    int jmax = 0;
    for (int i = 0; i <n; ++i)
    {
        if (arr[i][i] >max)
        {
            max = arr[i][i];
            imax = i;
            jmax = i;
        }
    }
    cout << "Maximum element is: " <<max<<" with coordinates:
"<<imax+1<<";"<<jmax+1;
return 0;
}
```

Input	Output
3 -1 5 6 2 -5 5 5 0 -6	Maximum element is: -1 with coordinates: 1;1

4 5 9 4 5 4 2 6 7 1 5 36 5 1 5 0 0	Maximum element is: 36 with coordinates: 3;3
3 5 6 7 2 5 6 1 2 3	Maximum element is: 5 with coordinates: 1;1

16. Secondary diagonal

The *secondary diagonal of a matrix* consists of those elements that lie on the diagonal that runs from top right to bottom left.

Return the sum of elements that lie on it.

Input

The first line contains integer n ($1 \leq n \leq 10^5$). Remaining lines contain integers — elements of matrix.

Output

Integer(sum).

Solution

```
#include<iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;
    int arr[n][n];
    int i,j,sum=0;
    for(i=0;i<n;i++)
    {
        for (j=0; j<n; j++)
        {
            cin>>arr[i][j];
        }
    }

    for(i=0;i<n;i++)
    {j=n-1-i; sum+=arr[i][j];}
    cout << sum;
    return 0;
}
```

}

Input	Output
3 -1 0 5 5 -2 5 -3 5 6	0
4 5 9 4 5 4 2 6 7 1 5 3 6 5 1 5 0 0	17

17. Christmas tree

Help Arman to print image of christmas tree in the screen.

Input

Given integer n ($1 \leq n \leq 10^5$), which is height of tree.

Output

Image of tree with stars.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        for(int j=i;j<n;j++){
            cout<<' ';
            for(int k=n;k>=n-(2*i);k--)
                cout<<'*';
            cout<<endl;
        }
    }
    return 0;
}
```

Input	Output
6	*

	<p>*** ***** ***** ***** *****</p>
5	<p>* *** ***** ***** *****</p>