

Assignment 1: 2d arrays

1. For an $n \times n$ matrix diagonal elements are given. All non-diagonal elements are equal, say x . Find out the minimum value of x such that the sum of diagonal elements is less than the sum of non-diagonal elements.

Input:

4
5 10 4 7

Output:

5 3 3 3
3 10 3 3
3 3 4 3
3 3 3 7

(Hint: Sum of diagonal element: 26, sum of other elements: 36)

2. In a matrix, calculate the sum of all neighboring elements of each diagonal element (up, down, left, right, and 4 diagonal elements -- a total of 8 elements). Print these values corresponding to each diagonal element. Also print the index of that diagonal element whose corresponding sum is highest.

Input:

5
3 2 0 4 5
1 10 4 -2 6
0 3 7 0 8
6 5 1 4 4
9 7 0 -1 3

Output:

3: 13
10: 20
7: 25
4: 22
3: 7
index: 3

3. Take a matrix of size $m \times n$. Find out its transpose using a function `transpose(arr, &m, &n)`.

Input:

2 3
1 1 1
2 2 2

Output:

1 2
1 2
1 2

4. Read a square matrix of size n. Do the following tasks:
- Print all unique values along each row.
 - Print all unique values along each column.
 - Print all unique values in the matrix.

Input:

```
4
5 1 2 1
4 10 3 5
2 1 4 4
1 2 0 7
```

Output:

Along rows:

```
1 2 5
3 4 5 10
1 2 4
0 1 2 7
```

Along columns:

```
1 2 4 5
1 2 10
0 2 3 4
14 5 7
```

Matrix:

```
0 1 2 3 4 5 7 10
```

5. Create an array of employee names.
- Arrange them in lexicographically sorted order
 - Print all unique names

Input:

```
8
Ram Mohan Shyam Amit Kritika Ram Mohit Amit
```

Output:

Sorted:

```
Amit Amit Kritika Mohan Mohit Ram Ram Shyam
```

Unique:

```
Amit Kritika Mohan Mohit Ram Shyam
```

6. Enter a square matrix and print the i^{th} row and j^{th} column whose sums are equal.

Input:

```
4
1 1 1 5
1 1 1 0
2 2 2 1
3 3 3 2
```

Output:

Row: 1

Column: 4

7. Check whether given strings are palindrome.

Input:

4
sos
abc
hello
abba

Output:

yes
no
no
yes

8. For an $n \times n$ matrix diagonal elements are given. Find if the matrix is such that diagonal element is equal to the sum of its neighboring (up/down/right/left only) elements.

Input:

5
3 2 0 4 5
1 10 4 -2 6
0 3 7 -1 8
6 5 1 4 5
9 7 2 -1 4

Output:

yes