1.Reverse Array bookmark_border

Max Score: 20

Print the array in reverse order.

Note:

Try solving this using recursion. Do not use any inbuilt functions / libraries for your main logic.

Input Format

The first line of input contains N - the size of the array and the second line contains the elements of the array.

Output Format

Print the given array in reverse order.

Constraints

1 <= N <= 100

0 <= ar[i] <= 1000

Example

Input

5

2 19 8 15 4

Output

4 15 8 19 2

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i,n;
    long long a[1000];
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        scanf("%lld",&a[i]);
    }
    for(i=n;i!=0;i--)
    {
        printf("%lld ",a[i]);
    }
    /* Enter your code here. Read input from STDIN. Print output to 
STDOUT */
    return 0;
}
```

2.Max Element in Array bookmark_border

Max Score: 20

Find the maximum element from the given array of integers.

Input Format

The first line of input contains N - the size of the array and the second line contains the elements of the array.

Output Format

Print the maximum element of the given array.

Constraints

1 <= N <= 10^3

$-10^9 <= ar[i] <= 10^9$

Example

Input

5

-2 -19 8 15 4

Output

15

Explanation

Self Explanatory

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i,n,j,temp=0;
    long long arr[1000];
    scanf("%d",&n);
    for(i=0;i<=n;i++)
    {
        scanf("%lld",&arr[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(arr[i]<arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    printf("%lld",arr[0]);
    return 0;
}
```

3.Complete the function *solveMeFirst* to compute the sum of two integers.

Example

Return .

Function Description

Complete the *solveMeFirst* function in the editor below.

solveMeFirst has the following parameters:

- *int a*: the first value
- *int b*: the second value

Returns

- *int*: the sum of and

Constraints

Sample Input

a = 2
b = 3

Sample Output

5

Code:

```
def solveMeFirst(a,b):
    # Hint: Type return a+b below
    return a+b

num1 = int(input())
num2 = int(input())
res = solveMeFirst(num1,num2)
print(res)
```

4.Given an array of integers, find the sum of its elements.

For example, if the array , , so return .

Function Description

Complete the *simpleArraySum* function in the editor below. It must return the sum of the array elements as an integer.

simpleArraySum has the following parameter(s):

- *ar*: an array of integers

Input Format

The first line contains an integer, , denoting the size of the array.

The second line contains space-separated integers representing the array's elements.

Constraints

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

6
1 2 3 4 10 11

Sample Output

31

```python
import math
import os
import random
import re
import sys

#
# Complete the 'simpleArraySum' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY ar as parameter.
#

def simpleArraySum(ar):
    return sum(ar)

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    ar_count = int(input().strip())

    ar = list(map(int, input().rstrip().split()))

    result = simpleArraySum(ar)

    fptr.write(str(result) + '\n')

    fptr.close()
```

5. Alice and Bob each created one problem for HackerRank. A reviewer rates the two challenges, awarding points on a scale from *1* to *100* for three categories: *problem clarity*, *originality*, and *difficulty*.

The rating for Alice's challenge is the triplet *a = (a[0], a[1], a[2])*, and the rating for Bob's challenge is the triplet *b = (b[0], b[1], b[2])*.

The task is to find their *comparison points* by comparing *a[0]* with *b[0]*, *a[1]* with *b[1]*, and *a[2]* with *b[2]*.

- If *a[i]* > *b[i]*, then Alice is awarded *1* point.
- If *a[i]* < *b[i]*, then Bob is awarded *1* point.
- If *a[i]* = *b[i]*, then neither person receives a point.

Comparison points is the total points a person earned.

Given *a* and *b*, determine their respective comparison points.

Example

a = [1, 2, 3]

b = [3, 2, 1]

- For elements *0*, Bob is awarded a point because *a[0] .*
- *For the equal elements a[1] and b[1], no points are earned.*
- *Finally, for elements 2, a[2] > b[2] so Alice receives a point.*

The return array is [1, 1] with Alice's score first and Bob's second.

Function Description

Complete the function compareTriplets in the editor below.

compareTriplets has the following parameter(s):

- *int a[3]: Alice's challenge rating*
- *int b[3]: Bob's challenge rating*

Return

- *int[2]*: Alice's score is in the first position, and Bob's score is in the second.

Input Format

The first line contains *3* space-separated integers, *a[0]*, *a[1]*, and *a[2]*, the respective values in triplet *a*.

The second line contains *3* space-separated integers, *b[0]*, *b[1]*, and *b[2]*, the respective values in triplet *b*.

Constraints

- *1 ≤ a[i] ≤ 100*
- *1 ≤ b[i] ≤ 100*

Sample Input 0

```
5 6 7
3 6 10
```

Sample Output 0

```
1 1
```

Explanation 0

In this example:

-
-

Now, let's compare each individual score:

- , so Alice receives point.
- , so nobody receives a point.
- , so Bob receives point.

Alice's comparison score is , and Bob's comparison score is . Thus, we return the array .

Sample Input 1

```
17 28 30
99 16 8
```

Sample Output 1

```
2 1
```

Code:

```python
import math
import os
import random
import re
import sys

#
# Complete the 'compareTriplets' function below.
#
# The function is expected to return an INTEGER_ARRAY.
# The function accepts following parameters:
#  1. INTEGER_ARRAY a
#  2. INTEGER_ARRAY b
#

def compareTriplets(a, b):
    # Write your code here
    aliceScore=0
    bobScore=0
    for i in range(len(a)):
        if a[i]==b[i]:
            continue
        else:
            if a[i]>b[i]:
                aliceScore=aliceScore+1
            else:
                bobScore=bobScore+1
    return [aliceScore,bobScore]


if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    a = list(map(int, input().rstrip().split()))

    b = list(map(int, input().rstrip().split()))

    result = compareTriplets(a, b)

    fptr.write(' '.join(map(str, result)))
    fptr.write('\n')

    fptr.close()
```

6. Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix is shown below:

1 2 3
4 5 6
9 8 9
The left-to-right diagonal = . The right to left diagonal = . Their absolute difference is .

Function description

Complete the function in the editor below.

diagonalDifference takes the following parameter:

- *int arr[n][m]*: an array of integers

Return

- *int*: the absolute diagonal difference

Input Format

The first line contains a single integer, , the number of rows and columns in the square matrix .

Each of the next lines describes a row, , and consists of space-separated integers .

Constraints

-

Output Format

Return the absolute difference between the sums of the matrix's two diagonals as a single integer.

Sample Input

3
11 2 4
4 5 6
10 8 -12

Sample Output

15

Explanation

The primary diagonal is:

```
11
  5
    -12
```

Sum across the primary diagonal: 11 + 5 - 12 = 4

The secondary diagonal is:

```
    4
  5
10
```

Sum across the secondary diagonal: 4 + 5 + 10 = 19

Difference: |4 - 19| = 15

Code:

```
import math
import os
import random
import re
import sys

#
# Complete the 'diagonalDifference' function below.
#
# The function is expected to return an INTEGER.
# The function accepts 2D_INTEGER_ARRAY arr as parameter.
#

def diagonalDifference(arr):
    # Write your code here
    arr_length=len(arr)
    primarySum=0
    secondarySum=0
    for i in range(arr_length):
        primarySum=primarySum+arr[i][i]
    for j in range(arr_length):
        secondarySum=secondarySum+arr[j][arr_length-1]
        arr_length=arr_length-1
    return abs(primarySum-secondarySum)
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
```

```
n = int(input().strip())

arr = []

for _ in range(n):
    arr.append(list(map(int, input().rstrip().split())))

result = diagonalDifference(arr)

fptr.write(str(result) + '\n')

fptr.close()
```