البدء بجمع عددين

لمعرفة كيفية المشاركة في المسابقات، شاهد الفيديو أدناه.

- مدة الزمن المحددة: 1 ثانية.
- حد الذاكرة : 256 ميجا بايت

في هذا السؤال نريد منكم ان تأخذو عددين طبيعيين من الإدخال الأساسي، ومن ثم تقوموا بطباعة حاصل جمع هذين العددين في المخرجات الرئيسية.

الإدخال

في السطر الأول من الإدخال، يتم كتابة عددين صحيحين موجبين a و b مفصولين بفاصلة واحدة.

$$1 \leq a, b \leq 100$$

الإخراج

في سطر الإخراج الوحيد، قم بطباعة قيمة $a + b$.

مثال

نموذج إدخال 1

```
3 5
```

نموذج إخراج 1

```
8
```

نموذج إدخال 2

```
1 1
```

نموذج إخراج 2

```
2
```

▼ طريقة الحل

```
a, b = map(int, input().split())
print(a+b)
```

▼ مثال على كود خاطئ

```
x = float(input('enter the first number '))
y =float(input('enter the second number '))
if x > 1 and x < 100:
    if y > 1 and y < 100:
        print(x + y)
```

الأخطاء:

١. المشكلة انه لايجب عرض رسائل للمستخدم مثل (enter the first number) .....

٢. داخل نص السؤال كتبنا انه يجب استخدام اعداد طبيعية اي (int) وفي الكود الخاص بك قمت باستخدام (float)

٣. ويجب تعريف المتغيرات x و y في سطر واحد وفي الكود الخاص بك قمت بتعرفيهن في سطرين

▼ جواب السؤال بلغات مختلفة

▼ Node.js

code.js

```
var readline = require('readline');
var rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout,
    terminal: false
});

var x, y;

rl.on('line', function (line) {

        var tmp = line.split(' ');
        x = parseInt(tmp[0]);
        y = parseInt(tmp[1]);
        var z = x + y;
        console.log(z);
    }
)
```

▼ JavaScript (V8)

code.js

```
var array = readline().split(' ');
var x = parseInt(array[0]);
var y = parseInt(array[1]);
console.log(x + y);
```

▼ Objective-C

code.m

```m
#import <stdio.h>

int main(void) {
    int n, m;
    scanf("%d %d", &n, &m);
    printf("%d" , n + m);
    return 0;
}
```

▼ Python2

code.py

```python
s = raw_input()
a, b = s.split(" ")
print int(a) + int(b)
```

▼ Python3

code.py

```python
s = input()
a, b = s.split(" ")
print(int(a) + int(b))
```

▼ C++

code.cpp

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

int main()
{
    int a, b;
    cin >> a >> b;
    cout << a + b << endl;

```

```
    return 0;
}
```

▼ C

code.c

```c
#include <stdio.h>

int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a + b);
    return 0;
}
```

▼ Java

Main.java

```java
import java.util.Scanner;
public class Main {
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(a + b);
    }
}
```

▼ Go

code.go

```go
package main

import "fmt"

func main() {
    var a , b int

```

```
    fmt.Scan(&a , &b)
    fmt.Println(a + b)
}
```

▼ C# Mono

C# code.cs

```
using System;

class Sum{
    public static void Main(){
        string[] s = Console.ReadLine().Split();
        int a = int.Parse(s[0]);
        int b = int.Parse(s[1]);
        Console.WriteLine(a + b);
    }
}
```

▼ Ruby

code.rb

```
inp = gets.split
a = inp[0].to_i
b = inp[1].to_i
puts (a + b)
```

▼ Perl

code.pl

```
my ($a, $b) = split ' ', <STDIN>;
print ($a + $b);
```

▼ PHP

code.php

```
<?php
$line = readline();
```

```php
[$a, $b] = explode(' ', $line);
echo $a + $b;
?>
```

▼ Swift 5

code.swift

```swift
// Swift 5.1
let line = readLine()!
let values = line.split(separator: " ");
let x = Int(values[0])!;
let y = Int(values[1])!;
print(x + y);
```

▼ Rust

code.rs

```rust
// Rust 1.61
use std::io::{self, BufRead};

fn main() {
    let mut line = String::new();
    let stdin = io::stdin();
    stdin.lock().read_line(&mut line).unwrap();
    line = line.trim().to_string();
    let numbers = line.split(" ").collect::<Vec<&str>>();
    let mut sum = 0;
    for i in 0..2 {
        let n = numbers[i].parse::<i32>().unwrap();
        sum += n
    }
    println!("{}", sum)
}
```

الدرجة إلى راديان

- مدة الزمان المحددة: 1 ثانية
- حد الذاكرة: 256 ميجابايت

يمكن حساب الزاوية عن طريق الدرجة وأيضاً عن طريق الراديان.

تُعطى لك الزاوية بالدرجة، ونريد منك ان تقوم بطباعتها بالراديان.

الإدخال

في سطر الإدخال الوحيد، يتم كتابة عدد صحيح وغير سالب d، والذي يمثل قيمة زاوية معطاة بالدرجات.

$$0 \leq d < 360$$

الإخراج

في سطر الإخراج الوحيد، يجب عليك طباعة عدد عشري يُمثل قيمة الزاوية المحددة بالراديان.

سيتم اعتبار إجابتك صحيحة فقط إذا كان الاختلاف بينها وبين الإجابة الصحيحة الفعلية أقل من 10^{-6}.

مثال

نموذج إدخال 1

```
180
```

نموذج إخراج 1

```
3.14159265
```

نموذج إدخال 2

```
58
```

نموذج إخراج 2

```
1.012290966
```

▼ طريقة الحل

```python
x = int(input())
result = (x*3.1415926535)/180
print(result)
```

▼ مثال على كود خاطئ

```python
x=int(input('enter the number'))
if 0 <= x <= 360:
    rst=(x*3.14)/180
    print('result :'+ str(rst))
```

الأخطاء:

١. المشكلة انه لايجب عرض رسائل للمستخدم مثل "enter the number" و "result :"

٢. في نص السؤال قيل انه يجب ان يكون التبديل 6 ارقام عشرية ولذلك يجب ان تستخدم القيمة 3.1415926535 بدلاً من 3.14

ترتيب السلسلة

- مدة الزمن المحددة: 1ثانية
- حد الذاكرة: 256 ميجابايت

يتم إعطاء سلسلة من الأعداد الصحيحة $a_1, a_2, \ldots, a_n$ لك. نريد منك أن تقوم بكتابة برنامج يقوم بفرز هذه السلسلة بترتيب تصاعدي ومن ثم طباعتها.

الإدخال

في السطر الأول من الإدخال، يتم كتابة عدد صحيح موجب n.

$$1 \leq n \leq 500\,000$$

في السطر التالي، يتم كتابة n عدد صحيح مفصولين بفاصلة واحدة بينهم. العدد i يُمثل قيمة a_i.

$$-10^9 \leq a_i \leq 10^9$$

الإخراج

في سطر الإخراج الوحيد، ستقوم بطباعة n أعداد صحيحة مفصولة بفاصلة واحدة بينها، وهي تمثل حالة السلسلة a بعد ترتيبها.

مثال

نموذج إدخال 1

```
5
3 6 2 1 2
```

نموذج إخراج 1

```
1 2 2 3 6
```

نموذج إدخال 2

```
3
3 2 1
```

نموذج إخراج 2

```
1 2 3
```

نموذج إدخال 3

```
4
17 -22 31 19
```

نموذج إخراج 3

```
-22 17 19 31
```

▼ طريقة الحل

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> numbers(n);
    for (int i = 0; i < n; i++) {
        cin >> numbers[i];
    }
    sort(numbers.begin(), numbers.end());

```

```
15
16      for (int i = 0; i < n; i++) {
17          cout << numbers[i] << " ";
18      }
19      return 0;
}
```

▼ مثال على كود خاطئ

```
1  x=int(input('enter the first number '))
2  y= input('enter the first number ')
3
4  if 1 <= x <= 500000:
5      y=y.split(' ')
6      if len(y) == x:
7          y = [int(num) for num in y]
8
9          m = sorted(y)
10
11         print(m)
```

الأخطاء:

١. مشكلة انه لايجب عرض رسائل للمستخدم مثل "enter the first number"

٢. يجب طباعة الاعداد داخل المصفوفة بشكل صحيح

اللعبة

- مدة الزمن المحدد: 1 ثانية
- حد الذاكرة: 256 ميجابايت

تعداد n عدد طبيعي مكتوبة على السبورة. أمير ومحمد يرغبان في إنشاء مصفوفة صحيحة منها.

في البداية، يختار أمير أكبر عدد على السبورة ويضعه في الخانة الأولى من المصفوفة ويقوم بحذف هذا العدد من السبورة (إذا كان هناك عدة نسخ من أكبر عدد على السبورة، يمكن لأمير اختيار أي منها بحرية). ثم يقوم محمد باختيار أصغر عدد على السبورة ويضعه في الخانة الثانية من المصفوفة ويقوم بحذف هذا العدد أيضًا (إذا كان هناك عدة نسخ من أصغر عدد على السبورة، يمكن لمحمد اختيار أي منها بحرية).

يقوم أمير بعد ذلك باختيار أكبر عدد متاح ووضعه في الخانة الثالثة من المصفوفة، وهكذا متواصلين بتكرار الخطوات السابقة حتى يتم إنشاء المصفوفة بالكامل (سيتم إنشاء المصفوفة عندما يتم حذف جميع الأرقام من السبورة).

الآن، المطلوب منكم طباعة المصفوفة النهائية وفقًا للطريقة المذكورة أعلاه.

الإدخال

في السطر الأول، يُعطى عدد n الذي يُمثّل عدد الأرقام الموجودة على السبورة. وفي السطر الثاني، يُعطى n عددًا طبيعيًا، والتي تُمثل الأرقام الموجودة على السبورة.

$$1 \leq n \leq 100$$

الأعداد التي على السبورة أقل من أو تساوي 100.

الإخراج

في سطر واحد، يجب عليك طباعة المصفوفة النهائية التي تم إنشاؤها بواسطة أمير ومحمد.

مثال

نموذج إدخال 1

```
7
2 5 2 7 1 6 4
```

نموذج إخراج 1

```
7 1 6 2 5 2 4
```

▼ طريقة الحل

```python
n = int(input())
numbers = list(map(int, input().split()))

matrix = [0] * n

numbers.sort(reverse=True)

for i in range(n):
    if i % 2 == 0:
        matrix[i] = numbers.pop(0)
    else:
        matrix[i] = numbers.pop()

for num in matrix:
    print(num, end=" ")
```

▼ مثال على كود خاطئ

```python
x=int(input('enter the first number '))
y= input('enter the first number ')

m=[]
if 1 <= x <= 100:
    y=y.split(' ')
```

```
    y = [int(num) for num in y]

    while len(m)-1 <= len(y):
        m.append(max(y))
        y.remove((max(y)))
        if len(m) < len(y):
            m.append(min(y))
            y.remove(min(y))
print(m)

```

الأخطاء:

١. مشكلة انه لايجب عرض رسائل للمستخدم مثل "enter the first number"

٢. يجب طباعة الاعداد داخل المصفوفة بشكل صحيح

٣. قم بتبديل الشرط while len(m) <= len(y) إلى هذا while len(y).

٤. قم بالتحقق من الكود الخاص بك مرة واحدة قبل الإرسال!!!

العدالة والمساواة

- مدة الزمن المحددة: 1ثانية
- حد الذاكرة: 256 ميجابايت

لمشاهدة مباراة كرة القدم، قام n شخصًا بتشكيل صف واحد خلف جدار الملعب. ارتفاع الشخص رقم i في هذا الصف هو h_i.

لضمان أن جميع هؤلاء الأشخاص يحصلون على رؤية أفضل للملعب، نرغب في وضع بعض الصناديق تحت أقدام هؤلاء الأشخاص بحيث يمكنهم رفع ارتفاعهم. يؤدي كل صندوق إلى زيادة ارتفاع شخص واحد بوحدة واحدة.

نقول أن العدالة تحققت عندما يكون اختلاف ارتفاع أي زوجين من الأشخاص الذين يشاهدون المباراة على الأقل d واحد.

نريد منك كتابة برنامج يحدد أدنى عدد من الصناديق التي يمكننا استخدامها لتحقيق العدالة.

الإدخال

في السطر الأول من الإدخال، يتم توفير عددين صحيحين و موجبين، n و d، وهما مفصولان بفراغ.

$$1 \leq n \leq 500\,000, \qquad 0 \leq d \leq 10^9$$

في n من السطور البعد، يتم اعطاء أعداد صحيحة $h_1, h_2, \ldots, h_n$ وقد تم فصلها بمسافة واحدة بينها.

$$1 \leq h_i \leq 10^9$$

الإخراج

في السطر الوحيد للمخرجات، يُطبع أدنى عدد من الصناديق اللازمة لتحقيق العدالة.

أمثلة

نموذج إدخال 1

```
3 1
1 2 8
```

نموذج إخراج 1

```
11
```

نموذج إدخال 2

```
4 0
1 5 3 6
```

نموذج إخراج 2

```
9
```

نموذج إدخال 3

```
1 3
5
```

نموذج إخراج 3

```
0
```

▼ طريقة الحل

```python
n, d = map(int, input().split(' '))
values = list(map(int, input().split(' ')))

max_value = max(values)
safe_value = max(0, max_value - d)
```

```
6
7  ans = 0
8  for value in values:
9      ans += max(0, safe_value - value)
10 print(ans)
```

▼ مثال على كود خاطئ

```
1  x,y = map(int, input().split())
2  heights = list(map(int, input().split()))
3
4  heights.sort()
5
6  b = 0
7  if len(heights) == x:
8      for i in heights:
9          diff = max(heights) - i
10
11         if diff >= y:
12             b += diff
13
14 print(b)
```

الأخطاء:

استخدم هذه الإدخالات في الكود الصحيح النتيجة يجب ان تكون 11 اما في هذا الكود الخاطئ تظهر النتيجة 13

```
3 1
1 2 8
```

معركة جوية

- مدة الزمن المحددة: 1ثانية
- حد الذاكرة: 256 ميجابايت

توجد مجموعة من الطائرات الحربية مصفوفة في صف وارتفاع كل واحدة منها مختلف عن الأخرى. كل طائرة حربية يمكنها استهداف الطائرات الحربية التي تقع أمامها فقط، وذلك بشرط أن يكون ارتفاع الطائرة المستهدفة أقل من ارتفاعها.

العدد الذي يمكن لطائرة حربية معينة استهدافه من الطائرات الحربية الأمامية يسمى **العدد الاستراتيجي**. على سبيل المثال، إذا كان بإمكان الطائرة الحربية ألف استهداف 3 طائرات حربية، فإننا نقول أن العدد الاستراتيجي للطائرة الحربية ألف هو 3.

يجب عليك حساب مجموع الأعداد الاستراتيجية لجميع الطائرات الحربية.

الإدخال

في سطر الإدخال الأول، يتم تقديم العدد n، الذي يُمثل عدد الطائرات الحربية. ثم في السطر التالي، يتم تقديم ارتفاع الطائرات الحربية الـ n بترتيب كسلسلة من الأعداد h_i.

$$1 \leq n \leq 100\ 000$$

$$1 \leq h_i \leq 100\ 000$$

الإخراج

في الإخراج، قم بطباعة مجموع الأعداد الأستراتيجة للطائرات.

مثال

نموذج إدخال 1

```
5
5 4 3 7 6
```

نموذج إخراج 1

```
4
```

▼ توضيح النموذج 1

الطائرة الحربية الأولى، التي يبلغ ارتفاعها 5، تقع في الخلف ولديها القدرة على استهداف الطائرتين الحربيتين الثانية والثالثة. وبالتالي، العدد الاستراتيجي لها هو 2. الطائرة الحربية الثانية يمكنها استهداف الطائرة الحربية الثالثة، والعدد الاستراتيجي لها هو 1. أما الطائرة الحربية الثالثة، فلا تستطيع استهداف الطائرتين الحربيتين الرابعة والخامسة بسبب ارتفاعها الأقل، لذا العدد الاستراتيجي لها هو 0. وبنفس الطريقة، العدد الاستراتيجي للطائرة الحربية الرابعة هو 1 والعدد الاستراتيجي للطائرة الحربية الخامسة هو 0.

بالتالي، مجموع أعداد الاستراتيجيات لجميع الطائرات الحربية سيكون 4.

نموذج إدخال 2

```
30
16 6 17 15 21 18 20 28 3 4 11 9 5 13 27 29 10 7 12 25 2 19 30 24 23 26 1 8
```

نموذج إخراج 2

```
202
```

▼ طريقة الحل

```python
def merge(left, right):
    merged = []
    inversions = 0
    i = j = 0


```

```python
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            j += 1
            inversions += len(left) - i

    merged.extend(left[i:])
    merged.extend(right[j:])

    return merged, inversions


def merge_sort(arr):
    if len(arr) <= 1:
        return arr, 0

    mid = len(arr) // 2
    left, inv_left = merge_sort(arr[:mid])
    right, inv_right = merge_sort(arr[mid:])
    merged, inversions = merge(left, right)

    return merged, inversions + inv_left + inv_right


n = int(input())
heights = list(map(int, input().split()))

sorted_heights, inversions = merge_sort(heights)
print(inversions)
```

النص التوضيحي للكود لحساب عدد الانعكاسات في القائمة باستخدام تقنية التقسيم والحل (Divide and Conquer) وبتعقيد زمني O(n log n):

هذا الكود يهدف إلى حساب عدد الانعكاسات في قائمة معينة من الأعداد. الانعكاس هو عبارة عن زوج من العناصر في القائمة حيث العنصر الأيمن أصغر من العنصر الأيسر. على سبيل المثال، في القائمة [5، 4، 3، 7، 6]، هناك أربع انعكاسات: (5، 4)، (5، 3)، (7، 6)، و (7، 6).

الخوارزمية:

الخوارزمية المستخدمة لحساب عدد الانعكاسات هي خوارزميةالتقسيم و الحل او فرق تسد. إليك كيف تعمل الخوارزمية:

١. **التقسيم:** تقسم القائمة إلى نصفين تقريبًا. يتم ذلك عن طريق تقسيم القائمة إلى جزئين متساويين (أو تقريباً متساويين) في كل تراكم.
٢. **الحل:** يتم حساب عدد الانعكاسات في الجزء الأيمن والجزء الأيسر من القائمة بشكل منفصل باستخدام الخوارزمية نفسها. ثم يتم دمج الجزئين وحساب عدد الانعكاسات الإضافية التي تحدث عند الدمج.
٣. **الدمج:** يتم دمج الجزئين الأيمن والأيسر لإنشاء قائمة مرتبة. خلال عملية الدمج، يتم حساب عدد الانعكاسات الإضافية التي تحدث عند دمج القائمتين.
٤. **الجمع:** يتم حساب إجمالي عدد الانعكاسات باجمع الانعكاسات من الجزئين الأيمن والأيسر والانعكاسات الإضافية التي تحدث أثناء الدمج.

المخرجات:

بعد تنفيذ الكود، سيتم طباعة إجمالي عدد الانعكاسات في القائمة المعطاة.

مثال:

في سياق مثال القائمة [5، 4، 3، 7، 6] السابق، ستكون الإجابة 4. يتم ذلك بواسطة تقسيم القائمة إلى نصفين: [5، 4، 3] و [7، 6]. بعد ذلك، يتم حساب الانعكاسات في النصفين الأيمن والأيسر بشكل منفصل، ومن ثم يتم دمج النصفين وحساب الانعكاسات الإضافية عند الدمج.