

**Corso di Laurea Magistrale in Ingegneria Informatica**

## **CORSO DI ALGORITMI E STRUTTURE DATI**

**Prof. ROBERTO PIETRANTUONO**

**Prima prova in itinere – secondo turno**

### **Indicazioni**

La consegna prevede un file "CognomeNome.PDF" in cui è riportato l'implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell'algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore  $O(f(n))$ ). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è  $O(n \log n)$  al netto della complessità dell'algoritmo  $x$ , che è non nota"). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

### **PROBLEMA 1**

Una sequenza di  $n > 0$  interi si dice *jolly* se i valori assoluti della differenza tra gli elementi successivi assumono tutti i valori da 1 a  $n - 1$ . Ad esempio,

1 4 2 3

è una sequenza *jolly*, perché le differenze assolute sono, rispettivamente 3, 2 e 1. La definizione implica che qualsiasi sequenza formata da un singolo intero è una sequenza *jolly*. Si implementi un programma per determinare se una data sequenza in ingresso è una sequenza *jolly*.

### **INPUT**

Ogni riga contiene un intero  $n \leq 3000$ , seguito da  $n$  interi che rappresentano la sequenza.

### **OUTPUT**

Per ogni riga di input, si generi una riga di output con il testo: "Jolly" o "Not jolly".

### **Sample Input**

4 1 4 2 3

5 1 4 2 -1 6

### **Sample Output**

Jolly

Not jolly

## PROBLEMA 2

Un gruppo di studenti si reca ogni anno in viaggio. Il gruppo si impegna in anticipo a condividere equamente le spese, ma non è pratico condividere ogni singola spesa nel momento in cui essa occorre. Quindi si accordano in modo che ciascuno di loro paghi per una determinata tipologia di spesa, ad esempio pasti, hotel, corse in taxi, biglietti aerei, ecc. Dopo il viaggio, si conteggiano le spese sostenute da ogni studente; essi si scambiano poi il denaro in modo tale che il costo netto per ciascuno studente sia lo stesso (con un errore di al più un centesimo). Si implementi un algoritmo che, a partire da un elenco di spese, calcoli la somma minima di denaro che deve essere scambiata per pareggiare (con un errore di al più un centesimo) tutte le spese degli studenti.

### INPUT

Lo *standard input* contiene le informazioni per diversi viaggi. Le informazioni per ogni viaggio sono costituite da una riga contenente un numero intero positivo,  $n$ , che è il numero di studenti in viaggio, seguita da  $n$  righe di input, ciascuna contenente l'importo, in euro e centesimi, speso da uno studente. Non ci sono più di 1000 studenti e nessuno ha speso più di € 10.000,00. Le informazioni dell'ultimo viaggio sono seguite da una riga contenente uno 0.

### OUTPUT

Per ogni viaggio, stampa una riga che indica l'importo totale di denaro, in euro e centesimi, che deve essere scambiato per pareggiare i costi degli studenti.

**Osservazione:** si osservi il secondo caso di test: il risultato è €11.99, e non €12.00 come si potrebbe pensare se lo studente che ha speso 15.01€ desse 6.00€ a chi ha speso €3.01, e chi ha speso €15.00 desse 6.00€ a chi ha speso €3.00.

### Sample Input

```
3
10.00
20.00
30.00
4
15.00
15.01
3.00
3.01
0
```

### Sample Output

```
€10.00
€11.99
```