

**Domanda 4**Risposta  
correttaPunteggio  
ottenuto 26,00  
su 26,00Contrassegna  
domanda

Sull'astronave Space Beagle è disposta una rete di sensori che registra la temperatura dell'aria. Gli  $N \times N$  sensori formano un quadrato di lato  $N=10$ . Ogni sensore è identificato da una coppia di coordinate intere  $x$  e  $y$ , dove  $x \in [0, N-1]$  e  $y \in [0, N-1]$ . Ogni sensore monitora i valori di temperatura con una frequenza non nota. Ogni volta che un sensore raccoglie una lettura, questa viene inviata all'elaboratore centrale dell'astronave e salvata in una riga di un file con il seguente formato:

```
ora x y temperatura
```

L'ora è una stringa nel formato "HH:MM:SS";  $x$  e  $y$  sono due numeri interi, mentre la temperatura (espressa in gradi Kelvin  $\geq 0$ ) è un numero reale. I 4 dati sono separati da spazi. Il file contiene dati riguardanti un'unica giornata in ordine cronologico. Ogni sensore ha letto almeno un dato nell'arco della giornata considerata.

Scrivere un programma che legga il file temp.txt contenente i dati di temperatura.

Il programma deve stampare a video, per ogni sensore, i dati medi di temperatura con una precisione di una cifra decimale, saltando eventuali coordinate per cui non ci siano misure. L'ordine di stampa non è rilevante. Inoltre, il programma deve stampare le coordinate del sensore che ha registrato il valore più alto e l'orario di tale dato.

Esempio: FILE INPUT:

temp.txt

```
01:10:00 1 1 7.1
01:10:02 0 1 7.2
01:10:03 1 1 8.1
01:10:05 0 0 7.3
01:10:15 1 0 7.4
01:10:18 1 0 7.6
```

OUTPUT:

```
Media temperature:
(0, 0) 7.3
(0, 1) 7.2
(1, 0) 7.5
(1, 1) 7.6
```

Il valore massimo è stato registrato dal sensore (1, 1) alle 01:10:03

**NOTA:** premete il pulsante "verifica risposta" per eseguire il codice (ricordatevi che non è necessario per l'esame che il codice sia compilabile ed eseguibile correttamente; la possibilità di eseguirlo vi permette di controllare se quello che avete scritto funziona. Verrà comunque corretta la logica del programma e l'approccio alla sua soluzione)

# LAIB 13A

## Laboratorio 13A

### Parte di Programmazione

Si vuole realizzare un programma Python che calcoli e stampi a video la quantità di caselle che sono state visitate da due robot.

Nel file 'traiettorie.txt' sono contenute le descrizioni dei percorsi seguiti da un insieme di micro-robot intelligenti liberi di muoversi nel piano discreto di dimensione  $N \times N$ .

Ciascuna riga del suddetto file descrive il percorso di uno (ed uno solo) micro-robot per punti adiacenti. Più precisamente, ciascuna riga specifica

- il nome del robot di massimo 10 caratteri
- la coordinata (x,y) del punto di partenza
- la sequenza di massimo 100 movimenti eseguiti dal micro-robot.

Tali movimenti sono limitati ad un incremento o decremento unitario in direzione orizzontale o verticale; nel file ciascun movimento è identificato da una coppia di simboli così codificati: il segno '+' identifica un incremento, il segno '-' identifica un decremento, il segno 'o' identifica uno spostamento orizzontale, il segno 'v' uno spostamento verticale. Si noti che la sequenza di simboli non contiene spazi.

Il seguente esempio fa riferimento a 2 righe del file 'traiettorie.txt' in cui i micro-robot, identificati dalla sigla "Rj6k" e "Ra9012" (da inserire nel codice come costanti), si muovono nello spazio  $5 \times 5$  seguendo la traiettoria rappresentata in figura (i punti di partenza sono alle coordinate (1,1) e (4,3)). La casella in alto a sinistra ha coordinate (1,1).

Rj6k 1 1				
x(1,1)		X	X	X
X			X	
X			X	
X	X	X	X	

Ra9012 4 3				
			X	
	X	X	X	
	X		x(4,3)	

Si noti che:

- uno stesso micro-robot può visitare la stessa posizione più volte (vedi esempio sopra)
- diversi micro-robot possono visitare le stesse posizioni
- i micro-robot possono seguire percorsi con lunghezze differenti
- tutte le traiettorie descritte sono contenute correttamente nel piano  $N \times N$ , non è dunque necessario verificarne la correttezza
- il numero di micro-robot, e quindi il numero di traiettorie memorizzate nel file, non è noto
- il formato del file è sempre corretto

Si scriva un programma utilizzi come costanti i seguenti due identificativi per i micro-robot da tracciare "Rj6k" e "Ra9012" e stampi a video la quantità di caselle che sono state visitate da entrambi i robot.

## Esempio

Come esempio si consideri il seguente file `traiettorie.txt` che descrive le traiettorie di diversi robot (nell'esempio 3):

```
Ra9012 4 3 -v-v+v-o-o+v  
Rj6k 1 1 +v+v+v+o+o+o-v-v-v-o+o+o  
Rq12a 2 2 -v-o+v+o-v+o+v-o
```

Il programma dovrà generare in output:

```
Ci sono 3 caselle visitate da entrambi i robot Rj6k e Ra9012
```

## Incendi1

Si realizzi un programma in Python che circoscriva un incendio boschivo assegnando una squadra di pompieri che ne controlli il perimetro. L'incendio è definito su una porzione quadrata e rappresentato su una matrice di caratteri di dimensioni note  $N \times N$  (con  $N$  costante nota a priori). Tale matrice è contenuta nel file chiamato "incendio.txt".

In un secondo file, chiamato "servizio.txt", sono elencati i pompieri in servizio, mentre in un terzo file, chiamato "riserva.txt", sono elencati i pompieri reperibili. Questi ultimi due file hanno lo stesso formato:

```
matricola;cognome;grado
```

dove **matricola** è un intero che rappresenta la matricola del pompiere, **cognome** è una stringa che rappresenta il cognome del pompiere e **grado** è un intero che rappresenta il grado di servizio del pompiere.

Ogni pompiere ha quindi un grado che ne identifica la capacità di contrastare le fiamme (i pompieri possono spegnere un numero di caselle pari al loro grado).

Il programma deve assegnare i pompieri sul perimetro dell'incendio per contrastare l'avanzata delle fiamme. Questa assegnazione viene fatta posizionando ogni pompiere nelle celle non incendiate prossime al perimetro delle fiamme. Una volta disposto, il pompiere spegne un numero di celle verticali pari al suo grado. L'assegnazione deve essere fatta partendo dai pompieri della lista in servizio. Solo in caso in cui il numero di pompieri non sia sufficiente, bisogna attingere dalla lista dei reperibili.

Si facciano le seguenti assunzioni:

1. In una matrice viene definito un singolo incendio
2. Il numero di pompieri (in servizio e reperibili) è sufficiente a coprire tutto il perimetro
3. Il nome del file viene inserito come costante nel programma

In caso in cui l'incendio sia completamente spento il programma deve comunicare al comando che l'incendio è sotto controllo. In caso in cui l'incendio sia ancora presente nel perimetro di competenza (definito dalla matrice) bisogna richiedere al comando l'intervento del Canadair. In entrambi i casi, il programma deve stampare a video la matrice di incendio aggiornata con le matricole dei pompieri disposte sul perimetro originale.

Esempi di matrice di incendio: 'f' indicano le zone incendiate, '.' indicano zone prive di incendio '+' indica incendio spento.

### Esempio di file "incendio\_1.txt"

```

. . . . f f f
. . . . f f f
. . . . f f f
. . . . f f f
. . . . f f f
. . . . f f f

```



deve stampare a video la matrice di incendio aggiornata con le matricole dei pompieri disposte sul perimetro originale.  
Esempi di matrice di incendio: 'f' indicano le zone incendiate, '.' indicano zone prive di incendio '+' indica incendio spento.

### Esempio di file "incendio\_1.txt"

```
. . . . f f f
. . . . f f f
. . . . . f f
. . . . . f f
. . . . . f f
. . . . .
. . . . .
. . . . .
```

### Esempio di file "incendio\_2.txt"

```
. . . . .
. . . . .
. f . . . .
. f f f . .
. f f f . .
. f f f f .
. f . . . .
. . . . .
```

### Esempio di file "servizio.txt"

```
1:Rossi;2
5:Bianchi;3
12:Verdi;3
```

### Esempio di file "riserva.txt"

```
7:Neri;2
23:Gialli;1
```

2:Vidua:2  
4:Viola:1

## Esempio di esecuzione del programma con il file "incendio\_1"

Incendio definito in incendio\_1.txt

```
. . . . + + f f
. . . . 1 + f f
. . . . . 5 + f
. . . . . . + +
. . . . . . + +
. . . . . . 12 7
. . . . . . .
. . . . . . .
```

Richiesta intervento Canadair

## Esempio di esecuzione del programma con il file "incendio\_2"

Incendio definito in incendio\_2.txt

```
. . . . . . .
. 1 . . . . .
. + 12 7 . . . .
. + + + . . . .
. + + + 2 . . . .
. + + + + . . . .
. + . 23 . . . .
. 5 . . . . .
```

Incendio sotto controllo

**i file specificati NON sono visibili direttamente, ma esistono e si possono aprire**

**NOTA:** premete il pulsante "verifica risposta" per eseguire il codice (ricordatevi che non è necessario per l'esame che il codice sia compilabile ed eseguibile correttamente; la possibilità di eseguirlo vi permette di controllare se quello che avete scritto funziona. Verrà comunque corretta la logica del programma e l'approccio alla sua soluzione)