

Esercizio filtro

Scrivere un programma che legga da **standard input** una stringa di caratteri in cui i caratteri corrispondenti a cifre decimali rappresentano dei numeri interi minori di 10.

All'interno della stringa letta, i caratteri corrispondenti a cifre decimali sono intervallati tra loro da sottostringhe formate da caratteri arbitrari.

Il programma deve considerare la sequenza di numeri interi nascosta all'interno della stringa letta da **standard input**.

Esempio:

All'interno della stringa di caratteri

```
&4&$4%mamma5!6mia6cosa1succede0
```

la sequenza di numeri interi nascosta è:

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

Il programma deve controllare che i numeri presenti all'interno della sequenza siano ordinati in senso decrescente, cioè dal più grande al più piccolo.

Nel caso in cui i numeri interi siano ordinati in senso decrescente, il programma deve stampare a video il messaggio `Sequenza nascosta ordinata.` . In caso contrario, il programma deve stampare a video il messaggio `Sequenza nascosta non ordinata.` .

Si assuma che:

- la stringa di caratteri letta da **standard input** sia unicamente definita da caratteri considerati all'interno dello standard US-ASCII;
- all'interno della stringa di caratteri letta da **standard input** non siano presenti i caratteri `-` `e` `+` `;`;
- all'interno della stringa di caratteri letta da **standard input** non siano presenti caratteri consecutivi corrispondenti a cifre decimali;
- chiaramente, all'interno della stringa di caratteri letta da **standard input** non sia presente nessun carattere di spaziatura, ossia un carattere il cui codice Unicode, passato come argomento alla funzione `func IsSpace(r rune) bool` del package `unicode` , fa restituire `true` alla funzione.

Esempio d'esecuzione:

```
$ go run esercizio_filtro.go
abc8def6asa2sd0
Sequenza nascosta ordinata.

$ go run esercizio_filtro.go
abc8d8e5aa4sd1e
Sequenza nascosta ordinata.

$ go run esercizio_filtro.go
abc9d8e6aa7sd1e
Sequenza nascosta non ordinata.
```

Test automatico:

L'esercizio filtro è considerato esatto **solo se** eseguendo il comando `go test` si ottiene il seguente output:

```
$ go test
PASS
ok
```

Invece, nel caso in cui l'output dovesse essere simile al seguente

```
$ go test
--- FAIL: TestFiltro (0.00s)
    esercizio_filtro_test.go:52:
        Input: <abc8d8e5aa4sd1e>
        Expected result: <Sequenza nascosta non ordinata.>
        Given result: <Sequenza nascosta ordinata.>

    esercizio_filtro_test.go:52:
        Input: <abc9d8e6aa7sd1e>
        Expected result: <Sequenza nascosta ordinata.>
        Given result: <Sequenza nascosta non ordinata.>
```

significa che almeno un caso tra quelli riportati nell'esempio d'esecuzione non è stato eseguito in modo corretto, ed il filtro è considerato **errato**.