

# Esercizio filtro

Scrivere un programma che legga da **standard input** una stringa di caratteri unicode senza spazi. Il programma deve stampare su standard output, una per riga, tutte le lettere dell'alfabeto inglese (nel set di caratteri ASCII) che sono seguite da una cifra decimale (nel set di caratteri ASCII).

**Esempio d'esecuzione:**

```
$ go run esercizio_filtro.go
albG93foEir92L
a
G
r

$ go run esercizio_filtro.go
E3e34$33ðñe3
e
e

$ go run esercizio_filtro.go
3eE31ld$4

$ go run esercizio_filtro.go
k10.e2uWe²23d
e
```

notare che in quest'ultimo caso 10. e ² **NON** sono cifre valide nel set di caratteri ASCII

**Test automatico:**

L'esercizio filtro è considerato esatto **solo se** segue le specifiche date e **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:40:
        ...
```

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**.

# Esercizio 1

Scrivere un programma che legga da **riga di comando** una stringa rappresentante una funzione quadratica nella forma  $c+bx+ax^2=0$ , con a,b,c interi e diversi da 0.

## Esempio:

$+2+1x+5x^2=0$

Il programma deve stampare su standard output quante e quali sono le soluzioni reali di questa equazione. Ogni soluzione deve essere approssimata usando 2 cifre decimali. Le soluzioni si possono trovare applicando la seguente formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Ne risulta che il numero di soluzioni reali totali di un'equazione quadratica possono essere zero, se il discriminante, ovvero la parte sotto la radice quadrata, è negativo; una, se il discriminante è nullo; oppure due altrimenti. Si ricorda che la funzione per calcolare la radice quadrata è disponibile nel package “math”.

## Esempio d'esecuzione:

```
$ go run esercizio_1.go +1+10x+4x^2=0
Ci sono due soluzioni reali: -0.10 e -2.40
```

```
$ go run esercizio_1.go +1-10x+4x^2=0
Ci sono due soluzioni reali: 2.40 e 0.10
```

```
$ go run esercizio_1.go +25+10x+1x^2=0
Esiste un'unica soluzione reale: -5.00
```

```
$ go run esercizio_1.go +2+1x+5x^2=0
Non ci sono soluzioni reali
```

# Esercizio 2

Uno dei compiti più importanti di un compilatore è quello di controllare se le parentesi sono ben bilanciate. Ad ogni parentesi aperta deve corrispondere una parentesi chiusa, e le coppie di parentesi devono essere innestate propriamente.

Un esempio di parentesi tonde ben bilanciate è il seguente: `(( ))()`

Un esempio di parentesi tonde *non* ben bilanciate è il seguente: `() ) (`

Notare che in quest'ultimo esempio, anche se ad ogni parentesi aperta corrisponde una parentesi chiusa, le coppie di parentesi non sono propriamente innestate (viene chiusa una parentesi di troppo ed una parentesi rimane aperta senza mai essere chiusa).

## Parte 1

Leggere da **standard input** una stringa. Se la stringa contiene caratteri diversi da parentesi tonda aperta ( e parentesi tonda chiusa ) (che ricordiamo sono caratteri ASCII) terminare l'esecuzione del programma, altrimenti stampare `bilanciata` se la stringa ha parentesi bilanciate o `non bilanciata` altrimenti. A tal fine implementare e usare una funzione `isBalanced(sequence string) bool` che riceve in input una stringa `sequence` composta di sole parentesi (aperte e chiuse) e che restituisce il valore booleano `true` se la stringa `sequence` è ben bilanciata, o `false` altrimenti.

## Parte 2

Stampare, in ordine crescente di lunghezza, tutte le sottostringhe ben bilanciate e il conteggio di quante volte appaiono nella stringa di input.

**Esempio d'esecuzione:**

```
$ go run esercizio_2.go
(( ))()
bilanciata
() 2
(( )) 1
(( ))() 1

$ go run esercizio_2.go
()
non bilanciata
() 1

$ go run esercizio_2.go
(( ))()
bilanciata
() 2
(( )) 1
(( ))() 1

$ go run esercizio_2.go
()()
non bilanciata
() 1
```

# Esercizio 3

Per svolgere le lezioni in presenza durante il COVID, dato il numero limitato di posti, gli studenti che intendono partecipare devono prenotarsi di volta in volta.

## Parte 1

Scrivere un programma per gestire le prenotazioni che:

- Legga la capienza massima di un'aula da riga di comando e la salvi come variabile intera.
- Legga da standard input una sequenza di righe di testo e termini la lettura quando, premendo la combinazione di tasti Ctrl+D, viene inserito da standard input l'indicatore End-Of-File (EOF).

Ciascuna riga definisce una lista di studenti che si sono prenotati per una lezione. La lista è in ordine di prenotazione, e può anche essere più piccola o più grande della capienza dell'aula. Le righe di testo sono composte di id studenti separati da spazio - si assumano tutti id validi e unici in una data riga. Un id è formato da <nome>.<cognome><numero> dove <numero> è un numero intero usato per per disambiguare studenti omonimi.

Esempio:

```
marco.rossi14 paolo.bianchi
```

Implementare una funzione `creaStudente(id string) (studente Studente)` che, data in ingresso la stringa id dello studente, restituisca una variabile di tipo `Studente` che contiene i valori id, nome e cognome dello studente, tutti di tipo stringa. Implementare una funzione `parseStudenti(s string) (listaPrenotazione []Studente)` che, data in ingresso una riga di testo contenente id degli studenti restituisca una slice di tipo `[]Studente` con tutte le variabili `Studente` corrispondenti a id degli studenti della riga. Implementare una funzione `printStudente(s Studente) string` che, data una variabile `Studente` restituisca la stringa `<nome> <cognome>`.

Esempio:

```
marco rossi
```

Implementare una funzione `printStudenti(lista []Studente)` che, data una slice di tipo `[]Studente` scriva su standard output la lista degli studenti nel formato `studenti: <nome1> <cognome1>, <nome2> <cognome2>...`

Esempio:

```
studenti: giorgia viola, francesco rossi
```

## Parte 2

Stabilire gli studenti abilitati a frequentare la lezione come segue: \* vengono abilitati a frequentare la lezione gli studenti fino alla capienza massima dell'aula \* hanno priorità: 1. gli studenti che hanno frequentato meno nelle lezioni precedenti 2. nell'ordine inverso in cui appaiono nella lista di prenotazione per la lezione stessa

Restituire, per ciascuna lezione, la lista degli studenti effettivamente abilitati a frequentare

Per test, si forniscono i file `esempio1.txt` ed `esempio2.txt` contenenti liste pregenerate, ad esempio:

```
$ cat esempio1.txt
andrea.celeste aurora.blu giorgia.viola3 giulia.bianchi5 francesco.rossi14
giulia.bianchi5 alessandro.rosa francesco.rossi14 lorenzo.neri2 giorgia.viola3 andrea.celeste
sofia.verdi1 lorenzo.neri2 giorgia.viola3
alessandro.rosa giulia.bianchi5 sofia.verdi1 giorgia.viola3 andrea.celeste francesco.rossi14 aurora.blu
aurora.blu giorgia.viola3 lorenzo.neri2 giulia.bianchi5 francesco.rossi14 alessandro.rosa sofia.verdi1 andrea.celeste
```

Esempio d'esecuzione:

```
$ go run esercizio_3.go 2 < esempio1.txt
studenti: francesco rossi, giulia bianchi
studenti: andrea celeste, giorgia viola
studenti: lorenzo neri, sofia verdi
studenti: aurora blu, alessandro rosa
studenti: andrea celeste, sofia verdi
```

```
$ go run esercizio_3.go 5 < esempio1.txt
studenti: francesco rossi, giulia bianchi, giorgia viola, aurora blu, andrea celeste
studenti: lorenzo neri, alessandro rosa, andrea celeste, giorgia viola, francesco rossi
studenti: sofia verdi, lorenzo neri, giorgia viola
studenti: aurora blu, sofia verdi, giulia bianchi, alessandro rosa, francesco rossi
studenti: andrea celeste, sofia verdi, alessandro rosa, giulia bianchi, lorenzo neri
```

```
$ go run esercizio_3.go 3 < esempio2.txt
studenti: giulia bianchi, giorgia viola, sofia verdi
studenti: francesco rossi, aurora blu, lorenzo neri
studenti: alessandro rosa, andrea celeste, giulia bianchi
studenti: alessandro rosa, aurora blu, sofia verdi
studenti: francesco rossi, andrea celeste, lorenzo neri
```

```
$ go run esercizio_3.go 6 < esempio2.txt
studenti: giulia bianchi, giorgia viola, sofia verdi, alessandro rosa, lorenzo neri
studenti: francesco rossi, aurora blu, andrea celeste, sofia verdi, giorgia viola, lorenzo neri
```

studenti: alessandro rosa, giulia bianchi, andrea celeste, francesco rossi, sofia verdi, giorgia viola  
studenti: aurora blu, giulia bianchi, alessandro rosa, francesco rossi, andrea celeste, lorenzo neri  
studenti: francesco rossi, andrea celeste, alessandro rosa, giulia bianchi, lorenzo neri, sofia verdi