

FILTRO: filtro.go

Scrivere un programma **filtro.go** che, data una sequenza di numeri interi positivi inseriti da linea di comando, stampi a schermo le potenze di 3, nell'ordine in cui appaiono.

Ad esempio:

```
$ go run filtro.go 1 9 2 27 3 25 6
1 9 27 3
```

DRAWX: drawX.go

Scrivere un programma **drawX.go** che stampi una X formata da asterischi la cui dimensione (altezza = larghezza) è passata da linea di comando (si assuma che sia un valore intero dispari).

Ad esempio:

```
$ go run drawX.go 7
*      *
 *    *
  *  *
   *
  * *
 *  *
*    *
*      *

$ go run drawX.go 1
*
```

MODA: moda.go

Nel file **moda.go** definire una funzione **moda** che prenda come argomento una slice di int e ritorni un'altra slice di int contenente i valori che compaiono con la massima frequenza nella slice passata come argomento.

Ad esempio:

```
s := []int{5, -2, 3, -2, 3, -2, 3}
fmt.Println(modas(s))
```

deve stampare:

```
[-2 3]
```

oppure:

```
[3 -2]
```

Solo ASCII: soloASCII.go

In un file **soloASCII.go** definire la funzione **soloASCII** che prenda come argomento una string e ritorni un'altra string composta unicamente dai caratteri ASCII presenti nella prima.

Ad esempio:

```
str := "⌘questa ée' una stringa 語formatàa di本 soli ASCII⌘\n"  
fmt.Print(soloASCII(str))
```

dovrebbe stampare (andando a capo):

```
questa e' una stringa formata di soli ASCII
```

DATA: data.go

In un file **data.go** definire:

- un tipo struttura **Data** contenenti i campi (di tipo uint) gg, mm, aa
- una funzione **creaData** che, dati tre valori uint passati come argomento, e corrispondenti nell'ordine a giorno, mese, anno, ritorni una coppia di valori (Data,bool) che rappresentino la Data creata in base ai valori passati e, a seconda che essi esprimano una data valida o (per qualsiasi motivo) no, true o false (in quest'ultimo caso la data ritornata dovrà essere {0,0,0}). Il controllo sull'anno bisestile è opzionale.
- una funzione **aggiorna** che, data una Data passata come argomento (per riferimento), che si assume valida, la aggiorni in modo che essa rappresenti la Data successiva (la funzione non dovrà ritornare nulla); nel caso di mancato controllo sull'anno bisestile, il 28 sarà l'ultimo giorno di Febbraio.
- una funzione **palindroma** che verifichi se una Data passata come argomento, che si assume valida, sia "palindroma" (ad esempio 02 02 2020 - si noti l'aggiunta di '0' prima dell'unità che indica, in questo caso, sia il giorno sia il mese), ritornando un valore bool.

Esempio d'uso:

```
var (
    g,m,a uint
    d Data
    ok bool
)

g, m, a = 1, 2, 2020

d,ok = creaData(g, m, a)

if ok {
    fmt.Println("creata:", d)
    aggiorna(&d)
    fmt.Println("data aggiornata:", d)

    if palindroma(d) {
        fmt.Println(d, "data palindroma")
    }
}
```