

filtro.go

Scrivete il programma `filtro.go` che, dato un numero di euro inserito da linea di comando, restituisca la combinazione di banconote e monete di taglio più alto possibile per dare la cifra specificata. Considerate i seguenti tagli (100, 50, 20, 10, 5, 2, 1).

Esempio d'esecuzione:

```
go run filtro.go 689
Taglio: 100 Quantità: 6
Taglio: 50 Quantità: 1
Taglio: 20 Quantità: 1
Taglio: 10 Quantità: 1
Taglio: 5 Quantità: 1
Taglio: 2 Quantità: 2
Taglio: 1 Quantità: 0
```

monovocale.go

Scrivete il programma `monovocale.go` che chieda all'utente di inserire da standard input una parola alla volta. Se la parola ha un solo tipo di vocale il programma deve scrivere su standard output "corretto", la vocale usata e quante volte è stata ripetuta (es: "corretto e 3"). Se la parola non ha un solo tipo di vocale il programma deve scrivere su standard output "sbagliato". Se la parola contiene altri caratteri oltre alle lettere minuscole "a-z" il programma deve scrivere su standard output "errore". Se la parola è "0" il programma termina.

Esempio d'esecuzione:

```
inserisci una parola
  banana
corretto a 3
inserisci una parola
  trapunta
sbagliato
inserisci una parola
  perchè?
errore
inserisci una parola
  0
```

space.go

Scrivete il programma "space.go" e definite un tipo (struttura) `Point3` che rappresenti punti dello spazio 3D mediante le tre coordinate `x`, `y` e `z`. Dovrà essere possibile, ad esempio, dichiarare:

```
p1 := Point3{1.3, 2.5, -2}
p2 := Point3{-3, -2.5, -2}
```

Definite (inferendone il prototipo) la funzione **proiezione** che dato un `Point3` pone a zero la coordinata `z` del punto. Ad esempio, data la precedente dichiarazione, la seguente sequenza:

```
proiezione(&p1)
fmt.Println(p1)
```

stampa:

```
{1.3 2.5 0}
```

Definite (inferendone il prototipo) la funzione **punti2D** che data una slice di punti restituisca una slice contenente solo i punti con la coordinata `z` uguale a `0`. Ad esempio, assumendo che `p1` e `p2` siano i punti precedenti, la sequenza:

```
s := []Point3{p1, p2, {}}
fmt.Println("punti con z 0:", punti2(s))
```

stampa:

```
punti con z 0: [{1.3 2.5 0} {0 0 0}]
```

eterogramma.go

Scrivete il programma "eterogramma.go" che, data una stringa inserita da linea di comando, verifichi se la stringa è un eterogramma (nessuna lettera ripetuta). Le frasi sono permesse e gli spazi non contano come lettere.

Esempio d'esecuzione:

```
go run eterogramma.go "The big dwarf only jumps"  
È un eterogramma
```

Esempio d'esecuzione 2:

```
go run eterogramma.go otorinolaringoiatra  
Non è un eterogramma
```

left_or_right.go

Scrivete il programma "left_or_right.go" che ogni turno scelga a caso destra o sinistra e stampi "Dove sto guardando?". L'utente può inserire d per destra e s per sinistra. Se l'utente indovina il lato il programma stampa: "Giusto, hai vinto!", altrimenti stampa: "Sbagliato, hai perso!". Se viene inserito "0" il gioco termina e stampa la percentuale di vittorie/sconfitte come segue:

```
Dove sto guardando?  
s  
Sbagliato, hai perso!  
Dove sto guardando?  
d  
Giusto, hai vinto!  
Dove sto guardando?  
0  
Vinto: 50% Perso: 50%
```