

Cifre

Scrivere un programma cifre.go dotato di

- una funzione

 contaCifre(s string) [10]int

che restituisce un array di lunghezza 10 che contiene il numero di

0, 1, 2, ... e 9 (in quest'ordine,) che si trovano nella stringa s.

Non ci sono vincoli sui tipi di caratteri in s.

- una funzione main() che legge una frase (tra "") da linea di comando che

contiene caratteri di qualsiasi tipo (cifre, lettere, simboli,

punteggiatura, lettere accentate, ecc.)

e stampa quante cifre di ogni tipo ci sono, ignorando gli eventuali altri caratteri.

Se manca l'argomento da linea di comando, deve stampare "manca argomento" e terminare.

Esempio

Input: "1, 2, 3; non c'è fiaba senza re; 1, 2, 3; venite giù da me; 4, 5, 6; siete dei babbei; 7,8,9; io sono già altrove."

Output: [0 2 2 2 1 1 1 1 1 1]

Lessico

Scrivere un programma lessico.go che

- stampa il seguente menu di opzioni

 + (Legge una riga e memorizza le parole)

 ? (Legge una parola e indica le righe che la contengono)

 p (Stampa le parole)

- legge stringhe da standard input

- il programma termina quando riceve un "end of file" (cioè EOF, pressione di 'CTRL-d')

Se la stringa inizia con:

- "+" (alimenta dizionario): il programma usa il rimanente della riga e memorizza in un "dizionario" le parole che la costituiscono;

- "?" (consulta dizionario): il programma usa il rimanente della riga e stampa i numeri di riga del dizionario in cui è comparsa la stringa;

- "p" (print): il programma stampa le parole presenti nel "dizionario", con l'elenco dei numeri di riga in cui compaiono;

Esempio

+ (Legge una riga e memorizza le parole)

? (Legge una parola e indica le righe che la contengono)

p (Stampa le parole)

+ la befana ha il fazzoletto e la gonna rattoppata

p

map[rattoppata:[1] la:[1 1] befana:[1] ha:[1] il:[1] fazzoletto:[1] e:[1] gonna:[1]]

+ ma quest'anno poverina la befana è raffreddata

? la

parola: la

righe [1 1 2]

? befana

parola: befana

```
righe [1 2]
? il
parola: il
righe [1]
p
map[raffreddata:[2] la:[1 1 2] il:[1] fazzoletto:[1] ma:[2] quest'anno:[2] poverina:[2] è:[2] befana:[1
2] ha:[1] e:[1] gonna:[1] rattoppata:[1]]
```

— — —

Percorso

Scrivere un programma percorso.go che ha:

- una struttura Punto con due campi float64 x, y che rappresentano le coordinate del punto nel piano cartesiano.

- una funzione

distanza(p1,p2 Punto) float64

che, dati due Punti p1 e p2, restituisce la distanza tra p1 e p2.

- una funzione

percorso(tappe []Punto) float64

che, data una sequenza di tappe, rappresentata da una slice di Punti, restituisce la distanza da percorrere, partendo dalla prima tappa, per tornare a questa, passando per tutte le altre tappe in ordine.

- una funzione

main()

che legge da standard input una sequenza (non vuota) di valori numerici (floating point) x1, y1, x2, y2, ..., xn, yn (per terminare la sequenza, dare invio e CTRL-D) e stampa la sequenza di punti letti e la distanza totale da percorrere per andare dal primo al secondo punto, dal secondo al terzo, ... e dall'ultimo al primo.

Nota : si puo` assumere, senza fare controlli, che la sequenza in input contenga almento i dati di un punto e che i dati siano nella forma attesa (float).

Esempi

Input: 0 3 0 0 4 0

Output:

tappe: [{0 3} {0 0} {4 0}]

percorso 12

Input: 1.5 3.2 5.1 7.6

Output:

tappe: [{1.5 3.2} {5.1 7.6}]

percorso 11.370136322841516

— — —

Filtro A

Scrivere un programma filtroA.go che legge una sequenza di valori interi da linea di comando e controlla che si alternino valori pari e valori dispari.

In questo caso il programma stampa il messaggio "sequenza valida", altrimenti "elemento in posizione i non valido", dove i è la posizione del primo elemento (da sinistra) che non rispetta la regola di alternanza o che non è un valore numerico.

In caso di mancanza di valori, il programma deve stampare "nessun valore in ingresso".

La sequenza può iniziare sia con un valore pari sia con uno dispari.

Si ricorda che lo zero è un numero pari.

Esempi

Input: 3 8 1 12

Output: sequenza valida

Input: 4

Output: sequenza valida

Input: 1 2 3 5

Output: elemento in posizione 4 non valido

Input: 1 2 3e4qw 5

Output: elemento in posizione 3 non valido

Input:

Output: nessun valore in ingresso

— — —

Filtro B

Scrivere un programma filtroB.go che legge una sequenza di valori interi da linea di comando e controlla che si alternino valori negativi e valori non negativi (positivi o zero).

In questo caso il programma stampa il messaggio "sequenza valida", altrimenti "elemento in posizione i non valido", dove i è la posizione del primo elemento (da sinistra) che non rispetta la regola di alternanza o che non è un valore numerico.

In caso di mancanza di valori, il programma deve stampare "nessun valore in ingresso".

La sequenza può iniziare sia con un valore positivo che con uno negativo.

Esempi

Input: -11 0 -6 8

Output: sequenza valida

Input: 4

Output: sequenza valida

Input: 0 -4 2 23 -12

Output: elemento in posizione 4 non valido

Input: 0 -4 2 2erere3 -12

Output: elemento in posizione 4 non valido

Input:

Output: nessun valore in ingresso