

Esercizio 1

Scrivere un programma che legga da **riga di comando** un numero intero `n` e da **standard input** una stringa di caratteri arbitrari in cui:

1) non è 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; 2) compaiono caratteri corrispondenti a cifre decimali; 3) i caratteri corrispondenti a cifre decimali sono intervallati tra loro da sottostringhe formate da caratteri arbitrari; 4) compare una e una sola volta il carattere `.`.

Il programma deve considerare il numero reale nascosto definito dalle cifre decimali e dal carattere `.` presenti all'interno della stringa letta da **standard input**.

Esempio:

All'interno della stringa di caratteri

```
&4&$4%mammas!6mia.6cosa1succede0
```

il numero reale nascosto è:

```
4456.610
```

Il programma deve stampare a video il valore del numero reale nascosto arrotondato alla `n`-esima cifra decimale.

Si assuma che, all'interno della stringa di caratteri letta da **standard input**:

- non siano presenti i caratteri `-` e `+`;
- il numero di caratteri corrispondenti a cifre decimali che compaiono dopo il carattere `.` sia strettamente maggiore di `n`.

Esempio d'esecuzione:

```
$ go run esercizio_1.go 3
&4&$4%mammas!6mia.6cosa1succede0
Numero nascosto: 445.6108
Numero arrotondato: 445.611

$ go run esercizio_1.go 2
&4&$4%mammas!6mia.6cosa1succede0
Numero nascosto: 445.6108
Numero arrotondato: 445.61

$ go run esercizio_1.go 3
L4b0r4t.0r10Pr06r4mmàz1on3
Numero nascosto: 404.016413
Numero arrotondato: 404.016

$ go run esercizio_1.go 2
L4b0r4t.0r10Pro6r4mmaz1on3
Numero nascosto: 404.016413
Numero arrotondato: 404.02

$ go run esercizio_1.go 0
&4&$4%mammas!6mia.6cosa1succede0
Numero nascosto: 445.6108
Numero arrotondato: 446

$ go run esercizio_1.go 1
Esam3.G05
Numero nascosto: 3.05
Numero arrotondato: 3.1
```


Esercizio 2

Definizione: I divisori propri di un numero naturale (un numero intero positivo) sono tutti i suoi divisori, tranne il numero stesso.

Definizione: Un numero naturale (un numero intero positivo) è perfetto se è uguale alla somma dei suoi divisori propri (per esempio, 6 è perfetto perché $6 = 1 + 2 + 3$).

Scrivere un programma che:

- legga da **riga di comando** due numeri interi positivi, rispettivamente `N` e `DIVISORIMIN` ;
- stampi a video tutte le coppie di interi positivi `a` e `b` , con `a <= N` e `b <= N` , tali che:
 - i. `a` e `b` abbiano al massimo 3 divisori propri in comune;
 - ii. `b` abbia un numero di divisori propri maggiore o uguale a `DIVISORIMIN` ;
 - iii. `a` sia un numero perfetto.

Oltre alla funzione `main()` , devono essere definite ed utilizzate almeno le seguenti funzioni:

- una funzione `DivisoriPropri(n int) []int` che riceve in input un valore `int` nel parametro `n` e restituisce un valore `[]int` in cui sono memorizzati tutti i divisori propri di `n` .

Si assuma che:

- i valori letti da **riga di comando** siano specificati nel formato corretto.

Esempio d'esecuzione:

```
$ go run esercizio_2.go 50 4
6 12
6 16
6 18
6 20
6 24
6 28
6 30
6 32
6 36
6 40
6 42
6 44
6 45
6 48
6 50
28 12
28 16
28 18
28 20
28 24
28 30
28 32
28 36
28 40
28 44
28 45
28 48
28 50

$ go run esercizio_2.go 20 2
6 4
6 6
6 8
6 9
6 10
6 12
6 14
```

6 15
6 16
6 18
6 20

\$ go run esercizio_2.go 20 3
6 6
6 8
6 10
6 12
6 14
6 15
6 16
6 18
6 20

\$ go run esercizio_2.go 189 10
6 60
6 72
6 84
6 90
6 96
6 108
6 120
6 126
6 132
6 140
6 144
6 150
6 156
6 160
6 168
6 180
28 60
28 72
28 90
28 96
28 108
28 120
28 132
28 144
28 150
28 156
28 160
28 180

\$ go run esercizio_2.go 30 4
6 12
6 16
6 18
6 20
6 24
6 28
6 30
28 12
28 16
28 18
28 20
28 24
28 30

Esercizio 3

Scrivere un programma che legga da **riga di comando** due stringhe di caratteri `Distretti.txt` e `Abbonati.txt`. Ciascuna delle stringhe è il nome di un file di testo memorizzato nella stessa directory in cui è memorizzato il programma.

a) Ogni riga contenuta nel file `Distretti.txt` è una stringa nel seguente formato:

nome;prefisso

La coppia di valori separati dal carattere `;` specifica un distretto telefonico relativamente alla rete telefonica italiana (rete di telefonia fissa):

1. *nome*: Una stringa che specifica il nome del distretto telefonico (ad es.: "Milano", "Roma", "Torino", ...).
2. *prefisso*: Una stringa interamente definita da caratteri corrispondenti a cifre decimali che specifica il prefisso associato al distretto telefonico.

Ogni distretto telefonico è identificato da un prefisso univoco formato da due, tre o quattro cifre decimali.

Si tenga infine presente che nel file `Distretti.txt` non è presente una riga per ciascuno dei distretti telefonici italiani, ma solo per alcuni di essi.

b) Ogni riga contenuta nel file `Abbonati.txt` è una stringa nel seguente formato:

cognome;nome;città;cap;via;numero_civico;numero_telefonico

Ogni n-pla di valori separati dal carattere `;` specifica un abbonato alla rete telefonica italiana (rete di telefonia fissa):

1. *cognome*: Una stringa che specifica il cognome dell'abbonato.
2. *nome*: Una stringa che specifica il nome dell'abbonato.
3. *città*: Una stringa che specifica la città di residenza dell'abbonato.
4. *cap*: Una stringa che specifica il C.A.P della città di residenza dell'abbonato.
5. *via*: Una stringa che specifica la via di residenza dell'abbonato.
6. *numero_civico*: Una stringa che specifica il numero civico di residenza dell'abbonato.
7. *numero_telefonico*: Una stringa, interamente definita da caratteri corrispondenti a cifre decimali, che specifica il numero telefonico dell'abbonato.

Ogni *numero_telefonico* è univoco, non può comparire in più di una riga del file `Abbonati.txt`.

In base al prefisso, ogni numero telefonico specificato nel file `Abbonati.txt` può essere associato o meno ad uno dei distretti telefonici specificati all'interno del file `Distretti.txt`.

Il programma deve:

- stampare a video il numero medio `M` di abbonati associati ad un distretto (cfr. **Esempio d'esecuzione**);
- stampare a video i distretti associati ad un numero di abbonati maggiore di `0` e minore di `M`, riportando per ogni distretto il numero di abbonati ad esso associati (cfr. **Esempio d'esecuzione**);
- stampare a video l'elenco degli abbonati relativi al distretto (specificato nel file `Distretti.txt`) a cui corrisponde il maggior numero di abbonati (cfr. **Esempio d'esecuzione**).

Oltre alla funzione `main()`, devono essere definite ed utilizzate almeno le seguenti funzioni:

- una funzione `StringIndirizzo(i Indirizzo) string` che riceve in input un'istanza del tipo `Indirizzo` nel parametro `i` e restituisce un valore `string` che corrisponde alla rappresentazione `string` di `i` nel formato `C.A.P.: CAP - CITTÀ\nVIA NUMERO_CIVICO`, dove `CAP`, `CITTÀ`, `VIA`, e `NUMERO_CIVICO` sono rispettivamente i valori string dei campi `cap`, `città`, `via`, e `numeroCivico` di `i`;

- una funzione `StringPersona(p Persona) string` che riceve in input un'istanza del tipo `Persona` nel parametro `p` e restituisce un valore `string` che corrisponde alla rappresentazione `string` di `p` nel formato `COGNOME NOME\nINDIRIZZO\nTelefono: NUMERO_TELEFONICO`, dove `COGNOME` e `NOME` sono i valori `string` dei campi `cognome` e `nome` di `p`, `INDIRIZZO` è la rappresentazione `string` del campo `indirizzo` (istanza del tipo `Indirizzo`) di `p`, e `NUMERO_TELEFONICO` è il valore `string` del campo `numeroTelefonico` di `p`.

Si assuma che:

- ogni riga del file `Distretti.txt` sia nel formato corretto;
- i valori presenti in ogni riga del file `Distretti.txt` specifichino correttamente un distretto telefonico;
- il file `Distretti.txt` non sia vuoto;
- i valori presenti in ogni riga del file `Abbonati.txt` specifichino correttamente un abbonato alla rete telefonica italiana;
- il file `Abbonati.txt` non sia vuoto.

Esempio d'esecuzione:

```
$ cat Distretti.txt
Milano;02
Brescia;030
Como;031
Varese;0332
Lecco;0341
Sondrio;0342
Bergamo;035
Lodi;0371
Cremona;0372
Mantova;0376
Pavia;0382

$ cat Abbonati.txt
Bianchi;Maria;Cremona;26010;Via Cavour;18;037211581249
Bianchi;Silvia;Lecco;23801;Via Matteotti;39;034111031796
Bianchi;Lorenzo;Pavia;27010;Via Mazzini;27;038415931720
Bianchi;Roberta;Como;22010;Via Garibaldi;22;03110431466
Verdi;Maria;Lecco;23801;Via Garibaldi;36;034112051762
Rossi;Silvia;Mantova;46010;Via Dante Alighieri;24;038613131232
Rossi;Mario;Varese;21010;Via Dante Alighieri;16;033211551851
Verdi;Mario;Lodi;26811;Via Dante Alighieri;23;037718161690
Rossi;Silvia;Bergamo;24121;Via Dante Alighieri;32;034614501189
...

$ go run esercizio_3.go Distretti.txt Abbonati.txt
Media abbonati per distretto: 1.636364
Abbonati per distretto:
Pavia -> 1
Varese -> 1
Lodi -> 1
Cremona -> 1
Milano -> 1
Como -> 1
4 abbonati distretto Mantova:
Verdi Roberta
C.A.P.: 46010 - Mantova
Via Cavour 26
Telefono: 037611861417
Rossi Mario
C.A.P.: 46010 - Mantova
Via Mazzini 35
Telefono: 037610311770
Verdi Silvia
C.A.P.: 46010 - Mantova
Via Dante Alighieri 10
Telefono: 037615231952
Rossi Mario
C.A.P.: 46010 - Mantova
```

Via Mazzini 35
Telefono: 037610311770