

## 1 Esercizio 1

Scrivere un programma "sniffer" per codici seriali che, dato un file di testo contenente un numero indefinito di caratteri, ricerchi stringhe nel seguente formato:

NNNN-NNNN-NNNN-NNNN

dove ogni carattere N corrisponde ad una cifra compresa tra 0 e 9 (0 e 9 inclusi).

Il programma dovrà poi stampare in un file in output il numero di stringhe compatibili con questo formato presenti nel file dato in input.

Procediamo ora con un esempio. Dato il file `input.txt` contenente le seguenti stringhe:

```
Spett.le ditta, vorrei comunicarvi i nuovi codici seriali per l'attivazione
del software da voi acquistato. Il primo codice è 1123-5813-2134-5589 mentre
il secondo codice è 1357-9111-3172-1232
```

se l'eseguibile è `a.out`, allora il comando:

```
./a.out input.txt output.txt
```

genererà un file chiamato `output.txt` che conterrà il seguente valore:

2

### Note

- Il programma dovrà prendere due argomenti come input: il nome del file con il testo da analizzare e il nome del file in cui salvare la risposta, in quest'ordine.
- I codici seriali sono composti solo da lettere corrispondenti a cifre e il delimitatore '-'.
- Per semplicità, assumete che ogni sequenza da individuare sia sempre preceduta e seguita da un separatore.
- Il file contiene stringhe con dimensione massima di 100 caratteri.

## 2 Esercizio 1

Scrivere un programma "sniffer" per codici seriali che, dato un file di testo contenente un numero indefinito di caratteri, ricerchi stringhe nel seguente formato:

`CCCC-CCCC-CCCC-CCCC`

dove ogni carattere C corrisponde ad una lettera compresa tra A e Z (A e Z inclusi).

Il programma dovrà poi stampare in un file in output il numero di stringhe compatibili con questo formato presenti nel file dato in input.

Procediamo ora con un esempio. Dato il file `input.txt` contenente le seguenti stringhe:

Spett.le ditta, vorrei comunicarvi i nuovi codici seriali per l'attivazione del software da voi acquistato. Il primo codice è KMKH-VQBL-YJXB-VYCQ mentre il secondo codice è VPBT-PZFD-YUKD-DGZY

se l'eseguibile è `a.out`, allora il comando:

```
./a.out input.txt output.txt
```

genererà un file chiamato `output.txt` che conterrà il seguente valore:

2

### Note

- Il programma dovrà prendere due argomenti come input: il nome del file con il testo da analizzare e il nome del file in cui salvare la risposta, in quest'ordine.
- I codici seriali sono composti solo da lettere maiuscole e il delimitatore '-'.
- Per semplicità, assumete che ogni sequenza da individuare sia sempre preceduta e seguita da un separatore.
- Il file contiene stringhe con dimensione massima di 100 caratteri.

### 3 Esercizio 1

Scrivere un programma "sniffer" per codici fiscali che, dato un file di testo contenente un numero indefinito di caratteri, ricerchi stringhe nel seguente formato:

```
CCCCNNNNNNNNNNNN
```

dove ogni carattere N corrisponde ad una cifra compresa tra 0 e 9 (0 e 9 inclusi), mentre ad ogni carattere C corrisponde una lettera tra A e Z (A e Z inclusi).

Il programma dovrà poi stampare in un file in output il numero di stringhe compatibili con questo formato presenti nel file dato in input.

Procediamo ora con un esempio. Dato il file `input.txt` contenente le seguenti stringhe:

```
Spett.le ditta di produzione software, vorrei acquistare presso  
di voi il vostro software a 359 euro. Per l'addebito, potete  
rilasciare fattura a Mario Rossi, con il seguente codice fiscale  
MMRS567891234567 oppure a Carla Bianchi, con il seguente  
codice fiscale CRBC001000110101
```

se l'eseguibile è `a.out`, allora il comando:

```
./a.out input.txt output.txt
```

genererà un file chiamato `output.txt` che conterrà il seguente valore:

2

#### Note

- Il programma dovrà prendere due argomenti come input: il nome del file con il testo da analizzare e il nome del file in cui salvare la risposta, in quest'ordine.
- Un codice fiscale è considerato valido se i suoi primi 4 caratteri sono lettere maiuscole e se i restanti 12 caratteri sono numeri.
- Per semplicità, assumete che ogni sequenza di 16 elementi da individuare sia sempre preceduta e seguita da un separatore.
- Il file contiene stringhe con dimensione massima di 100 caratteri.

## 4 Esercizio 1

Scrivere un programma "sniffer" per codici fiscali che, dato un file di testo contenente un numero indefinito di caratteri, ricerchi stringhe nel seguente formato:

```
NNNNNNNNNNNNCCCC
```

dove ogni carattere N corrisponde ad una cifra compresa tra 0 e 9 (0 e 9 inclusi), mentre ad ogni carattere C corrisponde una lettera tra A e Z (A e Z inclusi).

Il programma dovrà poi stampare in un file in output il numero di stringhe compatibili con questo formato presenti nel file dato in input.

Procediamo ora con un esempio. Dato il file `input.txt` contenente le seguenti stringhe:

```
Spett.le ditta di produzione software, vorrei acquistare presso  
di voi il vostro software a 359 euro. Per l'addebito, potete  
rilasciare fattura a Mario Rossi, con il seguente codice fiscale  
567891234567MMRS oppure a Carla Bianchi, con il seguente  
codice fiscale 001000110101CRBC
```

se l'eseguibile è `a.out`, allora il comando:

```
./a.out input.txt output.txt
```

genererà un file chiamato `output.txt` che conterrà il seguente valore:

2

### Note

- Il programma dovrà prendere due argomenti come input: il nome del file con il testo da analizzare e il nome del file in cui salvare la risposta, in quest'ordine.
- Un codice fiscale è considerato valido se i suoi ultimi 4 caratteri sono lettere maiuscole e se i restanti 12 caratteri sono numeri.
- Per semplicità, assumete che ogni sequenza di 16 elementi da individuare sia sempre preceduta e seguita da un separatore.
- Il file contiene stringhe con dimensione massima di 100 caratteri.