

# Esercitazione di Laboratorio 10

---

## Temi trattati

---

1. Lettura e scrittura di file
2. Gestione delle eccezioni
3. Elaborazione di dati

## Discussione

---

- A. Cosa succede se si cerca di aprire in lettura o in scrittura un file che non esiste?
- B. Qual è la differenza tra segnalare e gestire un'eccezione?
- C. Qual è la differenza tra accesso sequenziale e accesso casuale?

## Esercizi

---

### Parte 1 – Elaborazione di file

**Consegna:** per ciascuno degli esercizi seguenti, scrivere un programma in Python che risponda alle richieste indicate. Completare almeno un esercizio durante l'esercitazione, e i rimanenti a casa.

**10.1.1 Enola Gay.** Scrivere un programma che legga il file di testo `input.txt`, e scriva ogni riga letta nel file `output.txt` preceduta dal proprio numero di riga inserito come commento tra caratteri `'/*'` e `'*/'`. Se, ad esempio, il file `input.txt` contiene il testo:

```
Enola Gay
è il bombardiere che, il 6 agosto 1945,
sganciò su Hiroshima la prima bomba atomica
soprannominata Little Boy.
```

il file `output.txt` generato deve contenere il testo:

```
/*1*/Enola Gay
/*2*/è il bombardiere che, il 6 agosto 1945,
/*3*/sganciò su Hiroshima la prima bomba atomica
/*4*/soprannominata Little Boy.
```

[P7.2]

**10.1.2 Dal fondo.** Scrivere un programma che legga tutte le righe di un file di testo `input.txt`, ne inverta l'ordine e le scriva in un altro file `output.txt`. Ad esempio, se il file `input.txt` contiene il testo:

```
Mary had a little lamb
Its fleece was white as snow
And everywhere that Mary went
The lamb was sure to go.
```

il file `output.txt` generato deve contenere il testo:

```
The lamb was sure to go.
And everywhere that Mary went
```

Its fleece was white as snow  
Mary had a little lamb

[P7.9]

**10.1.3 Ricerca dell'anello.** Scrivere un programma che cerchi una data parola nel contenuto di un gruppo di file. Il programma deve chiedere all'utente di inserire in input:

- I. un elenco di **nomi di file** su una sola riga, separati da virgole;
- II. una parola da cercare.

I nomi dei file devono essere memorizzati in una lista, mentre la parola da cercare deve essere memorizzata in una variabile. Il programma deve visualizzare tutte le righe che contengono la parola, anche come sottostringa di altre parole, senza distinzione tra lettere maiuscole e minuscole. Ciascuna riga visualizzata deve essere preceduta dal nome del file in cui si trova. Ad esempio, se la parola da cercare è `'ring'`, e la lista contiene i file:

`book.txt`, `address.txt`, `homework.py`

allora il programma, elaborato il contenuto di tali file, deve visualizzare le righe dove si trova la parola da cercare con il seguente formato:

```
book.txt: There is only one Lord of the Ring, only one who can bend it to
his will
book.txt: The ring has awoken; it's heard its master's call.
address.txt: Kris Kringle, North Pole
address.txt: Homer Simpson, Springfield
homework.py: string = "text"
```

[P7.6]

**10.1.4 In albergo.** Il responsabile amministrativo di un albergo registra le vendite in un file di testo. Ogni riga contiene le seguenti 4 informazioni, separate da caratteri punto e virgola (';'):

- I. il nome del cliente;
- II. il servizio venduto;
- III. l'importo pagato;
- IV. la data dell'evento.

I servizi venduti possono essere, ad esempio, una cena, una conferenza, oppure un alloggio. Il formato corretto per questo file prevede che contenga 4 campi per riga, e che l'importo pagato contenga valori di tipo `float`. Scrivere un programma che legga questo file di testo, e visualizzi l'importo totale relativo a ciascun tipo di servizio, segnalando un errore se il file non esiste oppure se il suo formato non è corretto. [P7.29]

**10.1.5 Seconda possibilità.** Scrivere un programma che chiede all'utente di inserire una serie di valori di tipo `float`. Quando l'utente inserisce un valore che non è del tipo corretto, il programma deve fornire all'utente una seconda possibilità di inserire il valore, e dopo due possibilità deve interrompere la lettura dell'input, terminando immediatamente il programma. L'inserimento dei dati continua finché l'utente inserisce in input una stringa vuota (' '). Sommare tutti i valori correttamente specificati e visualizzare la loro somma quando l'utente ha finito di inserire i dati. Utilizzare la gestione delle eccezioni per rilevare gli input impropri. [P7.13]

## Parte 2 – Elaborazione di dati: corrispondenze e cifrature

**Consegna:** per ciascuno degli esercizi seguenti, scrivere un programma in Python che risponda alle richieste indicate. Completare almeno un esercizio durante l'esercitazione, e i rimanenti a casa.

**10.2.1 Cifratura monoalfabetica casuale.** Il cifrario di Cesare è un cifrario a sostituzione monoalfabetica, in cui ogni lettera del testo in chiaro è sostituita, nel testo cifrato, dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto. Poiché il numero di posizioni usato per la traslazione è fisso, il cifrario di Cesare è troppo facile da violare. Per rendere il testo cifrato più difficile da decifrare, una strategia alternativa è utilizzare come chiave, invece di un numero, una parola. Ad esempio, se la parola che si usa come chiave è FEATHER, per prima cosa si eliminano le lettere duplicate, ottenendo FEATHR, e poi si aggiungono in coda le rimanenti lettere dell'alfabeto, in ordine inverso. Si ottiene così la sequenza seguente.

F E A T H R Z Y X W V U S Q P O N M L K J I G D C B

Poi, si procede alla cifratura delle lettere, secondo lo schema seguente.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| F | E | A | T | H | R | Z | Y | X | W | V | U | S | Q | P | O | N | M | L | K | J | I | G | D | C | B |

Eventuali altri caratteri (ad esempio spazi, cifre, o segni di punteggiatura) devono rimanere invariati.

Scrivere un programma che richieda in input all'utente il nome di un file di testo da cifrare o decifrare, la scelta di quale operazione compiere (cifratura o decifratura), una parola chiave, e il nome di un file in cui scrivere l'output dell'elaborazione. Il programma deve memorizzare la parola chiave inserita dall'utente in una variabile, e utilizzarla per cifrare o decifrare il file di testo fornito in input, scrivendo poi il risultato delle elaborazioni nel file di output indicato. [P7.20]

**10.2.2 Libretto universitario.** Scrivere un programma che visualizzi l'elenco degli esami superati da uno studente, con i relativi voti. È disponibile un file `classes.txt`, che contiene i codici di tutti i corsi erogati nell'istituto scolastico (un college statunitense), il cui contenuto sarà simile a questo:

```
CSC1
CSC2
CSC46
CSC151
MTH121
...
```

Per ciascun corso, è disponibile un file il cui nome corrisponde al codice del corso, seguito da `'.txt'`. Questo file elenca le persone che hanno superato l'esame del corso, riportando per ciascuna il numero identificativo (`'Student ID'`) e il voto ottenuto. Ad esempio, il file `CSC2.txt` potrebbe contenere il testo:

```
11234 A-
12547 B
16753 B+
21886 C
...
```

Scrivere un programma che richieda all'utente di inserire in input uno 'Student ID', e visualizzi poi l'elenco degli esami superati ad esso associati, riportando i voti ottenuti per ciascun esame, rispettando il seguente formato.

Student ID: 16753

CSC2 B+  
MTH121 C+  
CHN1 A  
PHY50 A-  
[P7.28]

**10.2.3 Cifrario Playfair.** È possibile decifrare un testo mediante la semplice analisi della frequenza di occorrenza di singole lettere. Un modo per ostacolare questa strategia consiste nel cifrare in modo congiunto coppie di lettere. Il cifrario Playfair è un semplice schema che applica questa strategia. In questo schema, come prima cosa si sceglie una parola chiave, e si eliminano le lettere in essa duplicate. Poi si inserisce la parola chiave così elaborata in una scacchiera  $5 \times 5$ , seguita dalle rimanenti lettere dell'alfabeto inglese in ordine. Dato che ci sono soltanto 25 caselle e l'alfabeto inglese ha 26 lettere, la 'I' e la 'J' sono considerate indistinguibili. Verificare che nel testo sorgente le lettere siano pari, sennò aggiungere una 'Z'. Usando 'PLAYFAIR', trasformata in 'PLAYFIR', come parola chiave, si ottiene il seguente schema.

|   |   |   |   |   |
|---|---|---|---|---|
| P | L | A | Y | F |
| I | R | B | C | D |
| E | G | H | K | M |
| N | O | Q | S | T |
| U | V | W | X | Z |

Partendo da questa tabella, per cifrare una coppia di lettere, ad esempio 'AM', si individua la porzione di tabella che presenta 'A' ed 'M' a due estremi:

|   |   |   |   |   |
|---|---|---|---|---|
| P | L | A | Y | F |
| I | R | B | C | D |
| E | G | H | K | M |
| N | O | Q | S | T |
| U | V | W | X | Z |

La codifica della coppia 'AM' si ottiene trovando i valori ai due estremi restanti della porzione di tabella, in questo caso, 'FH'. Se le due lettere della coppia si trovano sulla stessa riga oppure sulla stessa colonna, come ad esempio, la coppia 'GO', la codifica si esegue scambiando le lettere tra

loro, ottenendo, in questo caso, 'OG'. La decifrazione segue le stesse regole. Scrivere un programma che cifri o decifri un file di testo secondo lo schema del cifrario Playfair. [P7.23]

**10.2.4 Legami covalenti.** Supponiamo che un file contenga le energie e le lunghezze di legame per dei legami covalenti, come quelli illustrati dalla tabella seguente.

| Bond (single  , double   , or triple    ) | Bond energy (kJ/mol) | Bond length (nm) |
|---|----------------------|------------------|
| C C                                       | 370                  | 0.154            |
| C  C                                      | 680                  | 0.13             |
| C   C                                     | 890                  | 0.12             |
| C H                                       | 435                  | 0.11             |
| C N                                       | 305                  | 0.15             |
| C O                                       | 360                  | 0.14             |
| C F                                       | 450                  | 0.14             |
| C Cl                                      | 340                  | 0.18             |
| O H                                       | 500                  | 0.10             |
| O O                                       | 220                  | 0.15             |
| O Si                                      | 375                  | 0.16             |
| N H                                       | 430                  | 0.10             |
| N O                                       | 250                  | 0.12             |
| F F                                       | 160                  | 0.14             |
| H H                                       | 435                  | 0.074            |

Il formato del file (denominato `bond_data.txt`) prevede che ogni riga in tabella corrisponda ad una riga di testo, e che, in ciascuna riga, i tre campi siano separati da uno spazio (' '), e non prevede l'intestazione. Ad esempio, un file che contiene i dati presentati in tabella potrebbe contenere il testo:

```
C|C 370 0.154
C||C 680 0.13
C|||C 890 0.12
...
```

Scrivere un programma che accetti in input un dato di una colonna e visualizzi i dati dalle altre due colonne nel file. Se il dato in ingresso ha una corrispondenza con più righe della tabella, il programma deve restituire i dati delle altre due colonne a tutte le righe che hanno una corrispondenza col valore in input. Ad esempio, una lunghezza di legame di `0.12` deve restituire sia il legame triplo `C|||C` e l'energia di legame `890 kJ/mol`, sia il legame singolo `N|O` e l'energia di legame `250 kJ/mol`. [P7.34]