

Fondamenti di Programmazione

A.A. 2020/2021, Appello del 22 Febbraio 2021

Il codice sorgente che contiene lo svolgimento della prova d'esame deve essere memorizzato in un singolo file (es: *cognome.c*). La consegna di tale codice sorgente avviene tramite la pagina del corso sul portale USienaIntegra: <https://elearning.unisi.it/course/view.php?id=4128>. E' necessario e sufficiente consegnare soltanto il file sorgente.

PROVA D'ESAME

Scrivere il programma che calcola il peso atomico di un composto.

Si supponga di fornire al programma un file di nome `tavola_elementi.txt` in cui ogni riga corrisponde ad un elemento ed è costituita da 4 campi (separati da spazi): il numero atomico, il nome, l'abbreviazione e il peso atomico dell'elemento. Ogni riga termina con un ritorno a capo.

Un esempio di file (ridotto rispetto al vero numero di elementi della tavola periodica) è il seguente:

```
1 Idrogeno H 1.0079
6 Carbonio C 12.0107
8 Ossigeno O 15.9994
16 Zolfo S 32.065
34 Selenio Se 78.96
52 Tellurio Te 127.6
```

Il programma chiede all'utente di inserire un composto e in risposta deve stampare su schermo il peso atomico del composto. Se, ad esempio, l'utente inserisce il seguente testo:

```
SeO3H
```

(il gruppo funzionale degli acidi selenonici: un atomo di selenio (Se), 3 di ossigeno (O3), e uno di idrogeno (H)), il programma, al termine della sua esecuzione, visualizzerà il messaggio:

```
Peso atomico del composto SeO3H : 127.966095
```

Per calcolare il peso atomico del composto inserito dall'utente il programma utilizza la tavola degli elementi memorizzata per determinare i pesi atomici degli atomi del composto; il peso atomico del composto è ottenuto come somma dei pesi degli atomi che compongono il composto, considerando la loro molteplicità. Nell'esempio precedente: $78.96 + 3 * 15.9994 + 1.0079 = 127.9661$

Si richiede che i dati di ciascun elemento vengano memorizzati ordinatamente nei campi di una struttura (`struct`) chiamati `num_atom`, `name`, `abbrev`, `mass_atom` (dei tipi opportuni). Si definisca un nuovo "tipo" associato alla struttura (nome del nuovo tipo: `elem`).

Il composto immesso dall'utente dovrà essere memorizzato in un array di caratteri `composto` di lunghezza massima 10 (definire una costante `LEN=10`). Si assume che: ogni elemento del composto è indicato tramite la sua abbreviazione, e un composto è una stringa priva di spazi contenente elementi e interi (supponiamo con valore nell'intervallo 2-9).

Nota: Le abbreviazioni di un elemento (es: Se, O, H) iniziano sempre con la maiuscola e sono costituite da uno o due caratteri. Un carattere ASCII che rappresenta un numero si converte nell'intero corrispondente sottraendo il codice ASCII di '0'. Per effettuare i controlli durante la scansione della stringa del composto potete usare le funzioni di libreria (in `ctype.h`), come ad esempio:

- `isalpha` (riceve un char e restituisce 0 se NON è una lettera dell'alfabeto)
- `isupper` (riceve un char e restituisce 0 se NON è una lettera maiuscola)
- `islower` (riceve un char e restituisce 0 se NON è una lettera minuscola)
- `isdigit` (riceve un char e restituisce 0 se NON è un numero)

Esercizio 1 [punti 8]

`main()`

Invoca le funzioni degli esercizi seguenti (in ordine).

Tutte le variabili passate come parametri alle funzioni devono essere dichiarate in `main`. Definire in `main` un array di N ($=6$ nell'esempio) elementi di tipo `elem` per memorizzare la tavola degli elementi da file. Per questo, il file `tavola_elementi.txt` viene aperto e poi chiuso in `main` dopo essere stato utilizzato. Se il file non esiste, il programma termina con un messaggio di errore di apertura del file, altrimenti, viene invocata la funzione dell'Esercizio 2 che carica il file. Se la funzione dell'Esercizio 2 ritorna -1 (errore di lettura), il programma deve terminare. Altrimenti, viene chiamata la funzione dell'Esercizio 3 che visualizza la tavola degli elementi. Il `main()` gestisce anche l'immissione da tastiera del composto come segue: richiede ripetutamente all'utente se vuole inserire un composto (risposte possibili `Y` o `N`); se l'utente digita `Y` richiede l'inserimento del composto, se digita `N` termina; infine, se digita qualunque altro carattere, avvisa l'utente di rispondere solo `Y` o `N`. Per processare il composto inserito dall'utente, viene chiamata la funzione dell'Esercizio 4 che si avvale della funzione dell'Esercizio 5 per la determinazione del peso atomico. Infine, viene stampato a video un messaggio col risultato del calcolo nel formato dell'esempio precedente.

Esercizio 2 [punti 6]

`load_elem(...)`

Carica i dati in memoria.

Legge il file `tavola_elementi.txt`, e popola un array composto di strutture di tipo `elem`. In particolare, riceve come parametri il puntatore al file, un array monodimensionale di strutture di tipo `elem` (parametro in uscita), un intero n corrispondente alla lunghezza dell'array (parametro in ingresso). È noto a priori che il file contiene n righe (corrispondenti al numero di elementi). Restituisce 0 se la lettura è avvenuta con successo, mentre restituisce -1 se il file contiene un formato errato (assumiamo che il formato sia errato se una riga contiene meno di 4 elementi del tipo corrispondente: la lettura si interrompe quando viene incontrata una riga non conforme).

Esercizio 3 [punti 5]

`print_table(...)`

Stampa a video la tabella degli elementi.

La funzione riceve come parametri in ingresso un array di strutture di tipo `elem` e la sua lunghezza n . Stampa a video il contenuto dell'array nel seguente formato (nel caso dell'esempio di file precedente):

Tavola degli elementi:

Numero atomico/Nome/abbreviazione/peso atomico

```
1/ Idrogeno/ H/ 1.007900
6/ Carbonio/ C/ 12.010700
8/ Ossigeno/ O/ 15.999400
16/ Zolfo/ S/ 32.064999
34/ Selenio/ Se/ 78.959999
52/ Tellurio/ Te/ 127.599998
```

Esercizio 4 [punti 8]

`float scan_request(...)`

Calcola il peso del composto.

Riceve come parametri in ingresso un array di caratteri per il composto, un array monodimensionale di strutture di tipo `elem`, e le loro lunghezze. La funzione scandisce la stringa del composto per identificare gli atomi e il loro numero. Una volta determinato un atomo e la sua quantità, viene chiamata la funzione dell'Esercizio 5 per determinare il suo peso atomico. Infine, il peso atomico trovato viene aggiunto alla somma parziale per il calcolo del peso atomico del composto. Il valore calcolato viene ritornato (`return`) a `main()`.

Esercizio 5 [punti 5]

`scan_table(...)`

Trova il peso atomico dell'elemento considerato.

Riceve come parametri in ingresso un array monodimensionale di caratteri per l'abbreviazione dell'elemento considerato, un array monodimensionale di tipo `elem` (per la tavola degli elementi) e la sua lunghezza. Cerca nel secondo array il peso atomico dell'elemento e la ritorna (`return`).