

# Fondamenti di Programmazione

A.A. 2020/2021

Appello del 22 Giugno 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>. È necessario e sufficiente consegnare soltanto il file sorgente.

---

## PROVA D'ESAME

Scrivere il programma che consente all'utente di comporre una frase selezionando le parole da un dizionario predefinito.

Il programma carica dal file `diz.txt` un elenco di parole che costituiranno il dizionario, disposte una per riga. Ad esempio,

```
dorme
gatto
in
mio
il
cantina
cane
un
vede
la
abbaia
```

Ogni parola è implicitamente associata ad un indice numerico in base alla sua posizione nel dizionario. Nel caso dell'esempio precedente, la parola con indice 0 è `dorme`, la parola con indice 1 è `gatto`, ..., la parola con indice 10 è `abbaia`.

Successivamente, il programma attende che l'utente inserisca da tastiera una sequenza di numeri interi non-negativi (un numero per volta, premendo invio subito dopo ciascun inserimento). La sequenza termina quando l'utente inserisce un numero minore di zero.

Il programma compone una frase selezionando le parole del dizionario che corrispondono agli indici inseriti dall'utente (escluso il numero negativo finale, ovviamente). Ad esempio, se l'utente inserisce

```
4 3 1 0 2 5 -1
```

la frase che viene composta sarà

```
il mio gatto dorme in cantina
```

---

Si richiede che ogni parola del dizionario venga memorizzata in una struttura di tipo `parola` (definire un nuovo tipo di dato), composta dal campo `testo`, che contiene la stringa associata alla parola, e dal campo `lunghezza`, che contiene la lunghezza della parola.

Nota: tutti gli array di questa prova si intendono di lunghezza massima pari a 50 (definire una costante intera `MAX = 50`).

### Esercizio 1 [punti 7]

`main()` *Invoca le funzioni degli esercizi seguenti (in ordine) e stampa a video alcuni risultati.*

Tutte le variabili passate come parametri alle funzioni degli esercizi seguenti devono essere dichiarate in `main`.

- Il file `diz.txt` viene aperto e poi chiuso in `main` dopo essere stato utilizzato. Se il file non esiste il programma termina, altrimenti, viene invocata la funzione dell'Esercizio 2 che carica in memoria il contenuto del file. Se la funzione dell'Esercizio 2 ritorna 0 (errore di lettura o file vuoto), il programma deve terminare stampando un messaggio di errore. Altrimenti, viene chiamata la funzione dell'Esercizio 3 che stampa a video il dizionario caricato.
- Successivamente, viene invocata la funzione dell'Esercizio 4, che attende che l'utente inserisca la sequenza di numeri e memorizza in un array le parole associate a tali numeri. Stampare in `main` la frase che si ottiene.
- La funzione dell'Esercizio 5, invocata alla fine, trova la parola più lunga della frase, che deve essere stampata in `main`.

### Esercizio 2 [punti 6]

`carica_dizionario(...)` *Carica il dizionario in memoria.*

Legge il file `diz.txt`, e popola un array di strutture di tipo `parola`. In particolare, riceve come parametri il puntatore al file, un array monodimensionale di strutture di tipo `parola` (parametro in uscita), un intero `n` corrispondente alla lunghezza della porzione effettivamente utilizzata dell'array (parametro in uscita). Restituisce 1 se è stata correttamente letta almeno una parola, altrimenti restituisce 0.

### Esercizio 3 [punti 6]

`stampa_dizionario(...)` *Stampa a video il dizionario caricato.*

La funzione riceve come parametri in ingresso un array di strutture di tipo `parola` (il dizionario) e la sua lunghezza effettiva `n`. Stampa a video il contenuto dell'array e il numero totale di caratteri nel dizionario, secondo il seguente formato (riferito all'esempio di pagina precedente)

```
>>> Dizionario:
0: dorme
1: gatto
2: in
3: mio
4: il
5: cantina
6: cane
7: un
8: vede
9: la
10: abbaia
>>> Totale caratteri: 42
```

### Esercizio 4 [punti 6]

`componi_frase(...)` *Colleziona le parole corrispondenti alla sequenza inserita dall'utente.*

Riceve come parametri in ingresso un array monodimensionale di strutture di tipo `parola` (il dizionario) e la sua lunghezza effettiva `n`. Riceve come parametri in uscita un array di strutture di tipo `parola` (la frase) e la sua lunghezza effettiva `m`. La funzione attende che l'utente inserisca un numero e memorizza in una posizione dell'array in uscita la parola del dizionario corrispondente al numero inserito. L'inserimento termina quando si inserisce un numero negativo, mentre i numeri  $\geq n$  vengono ignorati.

### Esercizio 5 [punti 7]

`trova_parola_lunga(...)` *Trova la parola più lunga della frase.*

Riceve come parametri in ingresso un array monodimensionale di strutture di tipo `parola` (la frase inserita dall'utente) e la sua lunghezza effettiva `m`. Restituisce un puntatore alla parola più lunga dell'array.