

Un `Dizionario` è una struttura dati che memorizza una lista di parole. Tutte le parole sono al più di 30 caratteri, e sono formate solo da lettere minuscole dell'alfabeto. Inoltre, non è possibile memorizzare in un `Dizionario` due o più parole identiche. Le operazioni che possono essere fatte su un `Dizionario` sono le seguenti:

✓ `inizializzaDizionario(D)` ; [2pt]

Funzione che inizializza il dizionario `D`. Inizialmente il dizionario non contiene alcuna parola.

✓ `inserisciParola(D, str)` ; [5pt]

Funzione booleana che aggiunge una nuova parola `str` nel dizionario `D`. Se l'inserimento ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

N.b. l'inserimento deve tener conto del formato richiesto nella funzione di stampa.

✓ `rimuoviParola(D, str)` ; [5pt]

Funzione booleana che rimuove la parola `str` dal dizionario `D`. Se l'eliminazione ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

✓ `contaIniziali(D, c)` ; [5pt]

Funzione che restituisce un numero intero. La funzione ritorna il numero delle parole nel dizionario `D` che iniziano con la lettera `c` passata per argomento alla funzione.

Esempio: se il dizionario contiene le parole `<zattera, botte, banca>` e la funzione viene chiamata con la lettera `'b'`, la funzione restituirà 2. Se viene chiamata invece con la lettera `'z'`, la funzione restituirà 1.

✓ `stampaDizionario(D)` ; [3pt]

Funzione che stampa a video la struttura dati `D` mostrando a video le parole in ordine di inserimento, dalla più recente alla meno recente, secondo il seguente formato:

`<zattera, botte, amo, frutta, banana, banca, farfalla>`

In questo esempio ci sono 7 parole, di cui `farfalla` è la parola inserita meno di recente nel dizionario, mentre `zattera` è la parola inserita più di recente. Le parole sono racchiuse fra parentesi angolate e sono separate da una virgola.

✓ `quanteOccorrenze(D)` ; [5pt]

Funzione che crea e ritorna un array di 26 interi `arr`, dove `arr[i]` memorizza quante parole iniziano con l'*i*-esima lettera dell'alfabeto. Assume che le lettere siano numerate a partire da 0.

Con riferimento al `Dizionario` mostrato nell'esempio della funzione `stampaDizionario`, l'array restituito dalla funzione è:

`[1, 3, 0, 0, 0, 2, 0, ... , 0, 0, 1]`, ovvero ci sono 1 parola che inizia con la lettera 'a', 3 parole che iniziano con 'b', 2 parole che iniziano con 'f', e 1 parola che inizia con 'z'.

Mediante il linguaggio C++, implementare il tipo `Dizionario` definito dalle precedenti specifiche utilizzando le **strutture**. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Esempio di funzione main()

```
int main( ){

    Dizionario D;
    inizializzaDizionario(D);

    inserisciParola(D, "amo");
    inserisciParola(D, "botte");
    inserisciParola(D, "zattera");
    inserisciParola(D, "botte");           //fallisce, già inserita

    stampaDizionario(D);                   // <zattera, botte, amo>

    inserisciParola(D, "ananas");
    rimuoviParola(D, "botte");

    stampaDizionario(D);                   // <ananas, zattera, amo>

    int n = contaIniziali(D, 'a');        // n = 2
    n = contaIniziali(D, 'b');           // n = 0

    inserisciParola(D, "botte");
    inserisciParola(D, "banana");
    inserisciParola(D, "banca");

    int *v = quanteOccorrenze(D);        // v = [2, 3, 0, .... , 0, 1]

    return 0;
}
```

Domande

1. **[2pt]** Rappresentare il numero in base 16 ABCDEF in base 8.
2. **[2pt]** Scrivere una funzione che apre un file di nome `str` (passato come argomento alla funzione) contenente una sequenza di numeri interi, li legge tutti e ne mostra a video la somma.
3. **[1pt]** Scrivere una funzione **ricorsiva** che prende in ingresso un array `arr` di dimensione `N` di numeri interi e che restituisce quanti numeri contenuti nell'array sono dispari.