`Classifica` mantiene le squadre ordinate per punteggio in ordine decrescente. Ogni squadra è identificata univocamente dal nome, che ha una massima lunghezza di 30 caratteri. Ad ogni squadra è associato un intero che corrisponde al punteggio. Implementare le seguenti operazioni che possono essere effettuate su `Classifica`:

✓ **`inizializzaClassifica(C)`** ; [3pt]
Funzione che inizializza una `Classifica` di squadre inizialmente vuota.

✓ **`aggiungiSquadre(C, str)`** ; [5pt]
Funzione che legge dal file di nome `str` una sequenza di squadre e le inserisce nella `Classifica` con punteggio associato 0. Se una squadra è già presente nella `Classifica` il suo punteggio rimane inalterato.

✓ **`RisultatoPartita(C, sq1, ris1, sq2, ris2)`** ; [5pt]
I possibili risultati di una partita sono: 1 per il pareggio, 3 per la vittoria, 0 per la sconfitta. Se entrambe le squadre sono presenti, vengono aggiornati i loro punteggi, altrimenti la funzione lascia la `Classifica` inalterata. Al termine della funzione la `Classifica` deve essere mantenuta ordinata per punteggio decrescente. Assumere che il punteggio sia inserito correttamente.

✓ **`stampaClassifica(C)`** ; [5pt]
Funzione che stampa a video nomi e punteggi delle squadre in `Classifica` con i 3 punteggi più alti. N.b. : Più squadre possono avere lo stesso punteggio, quindi può essere necessario stampare il nome di più di 3 squadre.

`Classifica`:

Benevento, 10

Entella, 10

Ascoli, 9

Pisa, 8

In questo esempio, il punteggio più alto è comune a due squadre, mentre solo una squadra ha punteggio 9 e solo una squadra ha punteggio 8.

✓ **`Punteggio(C)`** ; [4pt]
Funzione che ritorna il punteggio più comune in `Classifica`; se più punteggi sono presenti lo stesso numero di volte, restituisce il maggiore.

✓ **`cancellaUltime(C)`** ; [3pt]
Funzione che elimina dalla `Classifica` tutte le squadre con punteggio 0.

Mediante il linguaggio C++, implementare il tipo `Classifica` 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() {  
  
    Classifica C;  
    inizializzaClassifica(C);  
    stampaClassifica(C);           // Classifica:  
  
    aggiungiSquadre("squadre.txt");  
    stampaClassifica(C);  
  
    risultatoPartita(C, "Benevento", 3, "Entella", 0);  
    risultatoPartita(C, "Entella", 0, "Pisa", 3);  
    risultatoPartita(C, "Pisa", 3, "Benevento", 0);  
    stampaClassifica(C);  
  
    cout << Punteggio(C) << endl;    // 0  
  
    cancellaUltime(C);  
    stampaClassifica(C);  
  
    return 0;  
}
```

file "squadre.txt"

```
Benevento  
Entella  
Ascoli  
Pisa  
Perugia
```

```
/* Classifica:  
 * Benevento, 0  
 * Entella, 0  
 * Ascoli, 0  
 * Pisa, 0  
 * Perugia, 0  
 */  
  
/* Classifica:  
 * Pisa, 6  
 * Benevento, 3  
 * Entella, 0  
 * Ascoli, 0  
 * Perugia, 0  
 */  
  
/* Classifica:  
 * Pisa, 6  
 * Benevento, 3  
 */
```

Domande

1. [2pt] Rappresentare il numero ottale 321 in base 16.
2. [2pt] Scrivere una funzione che prende come argomenti una matrice quadrata di interi di dimensione $N \times N$. La funzione restituisce il `true` se la somma degli elementi sulla diagonale principale è uguale alla somma degli elementi sulla diagonale secondaria; altrimenti restituisce `false`.
3. [1pt] Scrivere una funzione **ricorsiva** che prende come argomento un array di interi e restituisce la somma degli elementi dell'array.