Un `NRistoranti` rappresenta il noto gioco di Alessandro Borghese in cui `N` giudici devono valutare `N` ristoranti, con `N` intero positivo strettamente maggiore di 1 e strettamente minore di 27. Il sistema deve quindi memorizzare la valutazione di ogni giudice per ogni ristorante. Ogni valutazione si compone di 4 voti: (i) location; (ii) servizio; (iii) menù; e (iv) conto. Ogni voto è un intero che deve essere compreso tra 0 e 10.

Dichiarare un `NRistoranti` **facendo un utilizzo efficiente della memoria** e implementare le seguenti operazioni che possono essere effettuate su `NRistoranti`:

--- Metodi invocati nella PRIMA PARTE di main.cpp: ---

✓ **`NRistoranti nr(N);`**

Costruttore che inizializza un oggetto `nr` di tipo `NRistoranti`, contenente le valutazioni di `N` giudici per `N` ristoranti. In caso il valore di `N` non sia valido, si assuma di default `N`=2. Ogni ristorante ha un nome assegnato, che è un carattere maiuscolo dell'alfabeto inglese che pertanto è uguale ad `'A'` per il primo ristorante, `'B'` per il secondo, e così via. Ad esempio, se `N`=4, i ristoranti si chiamano `'A'`, `'B'`, `'C'`, e `'D'`. I giudici non hanno un nome ma semplicemente un identificatore `k`=1,…,`N`. Il costruttore inizializza tutti i voti a zero.

✓ **`cout << nr;`**

Operatore di uscita per il tipo `NRistoranti`, che stampa a schermo le valutazioni su ogni ristorante da parte di ogni giudice, secondo il seguente formato:

```
2 ristoranti e 2 giudici
- Ristorante A, voto complessivo: 47
  Giudice 1:
    Location - 5
    Servizio - 6
    Menu - 7
    Conto - 6
  Giudice 2:
    Location - 4
    Servizio - 7
    Menu - 5
    Conto - 7
- Ristorante B, voto complessivo: 53
  Giudice 1:
    Location - 7
    Servizio - 7
    Menu - 8
    Conto - 7
  Giudice 2:
    Location - 5
    Servizio - 6
    Menu - 7
    Conto - 6
```

Come si nota dall'esempio, viene innanzitutto stampato il numero di ristoranti e giudici presenti in `nr`. Per ogni ristorante, viene stampato il voto complessivo, come somma di tutti i voti dati a quel ristorante. Vengono quindi stampate le valutazioni da parte di ogni giudice per ogni ristorante. I ristoranti sono stampati nello stesso ordine con cui sono stati inseriti. Per ogni ristorante, i giudici sono stampati in ordine crescente di identificatore. I 4 voti sono sempre stampati nel seguente ordine: location, servizio, menù, e conto. Si presti attenzione ai vari caratteri ',', ':', e '-' presenti. Si noti che dopo la riga di un ristorante, la riga di un giudice inizia con 2 caratteri spazio. Inoltre, le righe dei voti iniziano con 4 caratteri spazio.

- ✓ **`nr.aggiungiValutazione(nomeRistorante, giudice, location, servizio, menu, conto);`**

Aggiunge una nuova valutazione da parte del giudice avente identificatore `giudice`, per il ristorante avente nome `nomeRistorante`. Gli argomenti location, servizio, menu, e conto rappresentano il voto per ognuna delle voci. In caso di input invalidi, il metodo termina senza modificare l'oggetto `nr`. Per ogni coppia ristorante-giudice, la votazione può essere inserita una sola volta: qualunque ulteriore tentativo fallisce lasciando inalterato `nr`.

- ✓ **`nr.aggiungiBonus(nomeRistorante, bonus);`**

Dato il ristorante avente nome `nomeRistorante`, aggiunge un bonus alla valutazione di ogni giudice. Nello specifico, in ogni valutazione si individua il voto più basso e lo si incrementa del bonus, che è un intero compreso tra 1 e 10. Qualora ci fossero due o più voti con valore minimo, viene incrementato il primo nell'ordine (si ricordi che l'ordine è location, servizio, menu, conto). In caso di input non validi, il metodo fallisce ed `nr` rimane invariato. Nel caso in cui l'incremento portasse un voto ad essere maggiore di 10, quel voto non viene incrementato. Inoltre, questo metodo può essere applicato una sola volta al ristorante e può essere applicato solo ad un ristorante che ha già ricevuto le votazioni da tutti i giudici. In tutti gli altri casi, il metodo fallisce lasciando `nr` inalterato.

--- Metodi invocati nella SECONDA PARTE di main.cpp: ---

- ✓ **`NRistoranti nr2 = nr;`**

Costruttore di copia.

- ✓ **`~NRistoranti();`**

Qualora sia necessario, implementare il distruttore.

- ✓ **`~nr;`**

Operatore di complemento su `nr`, che modifica `nr` riordinando i ristoranti in ordine decrescente di valutazione ottenuta. Nello specifico, per ogni ristorante devono essere sommati tutti i voti assegnati da tutti i giudici. Bisogna quindi ordinare i ristoranti in base a questo voto complessivo. Nel caso vi fossero due ristoranti con stesso voto complessivo, l'ordinamento viene fatto sul nome del ristorante e considerando l'ordine inverso a quello alfabetico. Ad esempio, qualora `'A'` e `'B'` avessero entrambi un voto complessivo pari a 64, l'ordinamento sarebbe `'B'` seguito da `'A'`. Il metodo non inverte l'ordine dei giudici. Il metodo restituisce un riferimento all'oggetto `nr` modificato.

- ✓ **`!nr;`**

Operatore di negazione logica su `nr`, che crea un nuovo oggetto `NRistoranti`. Il nuovo oggetto contiene solo i ristoranti --- e le relative valutazioni da parte di tutti i giudici --- che hanno ricevuto un voto

complessivo pari o superiore a 100. I ristoranti che rispettano questo criterio vengono riportati nel nuovo oggetto nello stesso ordine con cui sono presenti in `nr`.

Mediante il linguaggio C++, realizzare il tipo di dato astratto **`NRistoranti`**, definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc. **Gestire le eventuali situazioni di errore**.

Uscita che deve produrre il programma

```
--- PRIMA PARTE ---
Test costruttore e operatore di stampa:

3 ristoranti e 3 giudici
- Ristorante A, voto complessivo: 0
  Giudice 1:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 2:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 3:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
- Ristorante B, voto complessivo: 0
  Giudice 1:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 2:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 3:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
- Ristorante C, voto complessivo: 0
  Giudice 1:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 2:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0
  Giudice 3:
    Location - 0
    Servizio - 0
    Menu - 0
    Conto - 0

Test funzione aggiungiValutazione:

3 ristoranti e 3 giudici
- Ristorante A, voto complessivo: 83
  Giudice 1:
    Location - 7
    Servizio - 6
    Menu - 7
    Conto - 7
```

```
  Giudice 2:
    Location - 5
    Servizio - 7
    Menu - 5
    Conto - 7
  Giudice 3:
    Location - 8
    Servizio - 8
    Menu - 8
    Conto - 8
- Ristorante B, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 7
    Menu - 8
    Conto - 7
  Giudice 2:
    Location - 7
    Servizio - 8
    Menu - 7
    Conto - 6
  Giudice 3:
    Location - 8
    Servizio - 7
    Menu - 8
    Conto - 9
- Ristorante C, voto complessivo: 112
  Giudice 1:
    Location - 10
    Servizio - 10
    Menu - 10
    Conto - 10
  Giudice 2:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
  Giudice 3:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9

Test funzione aggiungiBonus:

3 ristoranti e 3 giudici
- Ristorante A, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 9
    Menu - 7
    Conto - 7
  Giudice 2:
    Location - 8
    Servizio - 7
    Menu - 5
    Conto - 7
  Giudice 3:
    Location - 8
    Servizio - 8
    Menu - 8
    Conto - 8
- Ristorante B, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 7
    Menu - 8
    Conto - 7
  Giudice 2:
    Location - 7
    Servizio - 8
    Menu - 7
    Conto - 6
  Giudice 3:
    Location - 8
```

```
    Servizio - 7
    Menu - 8
    Conto - 9
- Ristorante C, voto complessivo: 112
  Giudice 1:
    Location - 10
    Servizio - 10
    Menu - 10
    Conto - 10
  Giudice 2:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
  Giudice 3:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9

--- SECONDA PARTE ---
Test costruttore di copia:
Oggetto copiato:

3 ristoranti e 3 giudici
- Ristorante A, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 9
    Menu - 7
    Conto - 7
  Giudice 2:
    Location - 8
    Servizio - 7
    Menu - 5
    Conto - 7
  Giudice 3:
    Location - 8
    Servizio - 8
    Menu - 8
    Conto - 8
- Ristorante B, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 7
    Menu - 8
    Conto - 7
  Giudice 2:
    Location - 7
    Servizio - 8
    Menu - 7
    Conto - 6
  Giudice 3:
    Location - 8
    Servizio - 7
    Menu - 8
    Conto - 9
- Ristorante C, voto complessivo: 112
  Giudice 1:
    Location - 10
    Servizio - 10
    Menu - 10
    Conto - 10
  Giudice 2:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
  Giudice 3:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9

Test eventuale distruttore:
```

```
Distruttore chiamato

Test operatore ~ (ordinamento):

3 ristoranti e 3 giudici
- Ristorante C, voto complessivo: 112
  Giudice 1:
    Location - 10
    Servizio - 10
    Menu - 10
    Conto - 10
  Giudice 2:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
  Giudice 3:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
- Ristorante B, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 7
    Menu - 8
    Conto - 7
  Giudice 2:
    Location - 7
    Servizio - 8
    Menu - 7
    Conto - 6
  Giudice 3:
    Location - 8
    Servizio - 7
    Menu - 8
    Conto - 9
- Ristorante A, voto complessivo: 89
  Giudice 1:
    Location - 7
    Servizio - 9
    Menu - 7
    Conto - 7
  Giudice 2:
    Location - 8
    Servizio - 7
    Menu - 5
    Conto - 7
  Giudice 3:
    Location - 8
    Servizio - 8
    Menu - 8
    Conto - 8

Test operatore ! (filtro >= 100):

1 ristoranti e 3 giudici
- Ristorante C, voto complessivo: 112
  Giudice 1:
    Location - 10
    Servizio - 10
    Menu - 10
    Conto - 10
  Giudice 2:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
  Giudice 3:
    Location - 9
    Servizio - 9
    Menu - 9
    Conto - 9
```

Note per la consegna:

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In questo caso, i docenti procederanno alla valutazione dell'elaborato **solo se** lo studente avrà completato l'autocorrezione del proprio elaborato.
In **tutti** gli altri casi (per esempio, il programma non compila, non collega, non esegue o la prima parte dell'output non coincide con quella attesa), l'elaborato è considerato **insufficiente** e, pertanto, **non verrà corretto**.