

Una CasaDiCura rappresenta un sistema per gestire gli ospiti all'interno di una casa di cura ed i farmaci che questi devono assumere. Una CasaDiCura è caratterizzata da un nome, che è una stringa non vuota di al più 20 caratteri. Gli ospiti sono univocamente individuati attraverso identificativi interi crescenti che partono da 1. Ogni ospite ha un insieme di farmaci da assumere. Ogni farmaco è caratterizzato da: i) un nome, che è una stringa non vuota di al più 15 caratteri; ii) un attributo che chiarisce se il farmaco è stato assunto o meno; e iii) un intero non negativo (con valori compresi tra 0 e 23) che esplicita a quale ora deve essere assunto il farmaco.

Implementare le seguenti operazioni che possono essere effettuate su CasaDiCura:

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

✓ **CasaDiCura cc(nome, numeroOspiti);**

Costruttore che inizializza una CasaDiCura caratterizzata da `nome` e contenente un numero di ospiti pari a `numeroOspiti`. All'inizio, ogni ospite ha un insieme vuoto di farmaci.

✓ **cout << cc;**

Operatore di uscita per il tipo CasaDiCura, che stampa a schermo tutti gli ospiti ed i farmaci associati, nel seguente formato:

Casa di Cura 'Villa Giulia'

```
- Ospite 1:           (questa riga inizia con 0 spazi bianchi, ossia in prima colonna)
  1. 'Tachipirina'   (questa riga inizia con 2 spazi bianchi)
    Assunto          (questa riga inizia con 5 spazi bianchi)
    Ore 8            (anche questa riga inizia con 5 spazi bianchi)
  2. 'Pantoprazolo' (questa riga torna ad iniziare con 2 spazi bianchi)
    Non assunto
    Ore 19
- Ospite 2:           (questa riga inizia con 0 spazi bianchi, ossia in prima colonna)
  1. 'Voltaren'
    Non assunto
    Ore 12
```

Come si nota dall'esempio, viene innanzitutto stampato il nome dell'oggetto `cc`, tra singoli apici. Poi, per ogni ospite, viene stampata la stringa 'Ospite' seguita dall'identificativo e, a seguire, l'elenco numerato dei farmaci associati all'ospite. Per ogni farmaco, viene stampato: i) il nome tra singoli apici; ii) l'informazione se il farmaco è stato o non è stato assunto; iii) l'orario in cui effettuare l'assunzione. Si noti che gli ospiti sono stampati in ordine crescente di identificativo. Inoltre, per ogni ospite, l'elenco dei suoi farmaci è stampato in base all'orario crescente di assunzione. Si noti anche che, rispetto alla riga di un ospite, la riga contenente il nome del farmaco inizia dopo due caratteri spazio, mentre le righe contenenti le altre informazioni su un farmaco iniziano dopo cinque caratteri spazio.

✓ **cc.aggiungiFarmaco(idOspite, nome, orario);**

Per l'ospite avente `idOspite`, aggiunge un nuovo farmaco avente `nome`, il quale deve essere assunto ad `orario`. Inizialmente, il farmaco risulta come non assunto. Per un dato ospite, non possono esistere due farmaci aventi stesso nome o stesso `orario`. Il metodo restituisce `true` in caso di aggiunta con successo. Altrimenti, lascia `cc` invariato e restituisce `false`.

✓ `cc.assumiFarmaci(idOspite, orario);`

Per l'ospite avente `idOspite`, effettua l'assunzione del farmaco previsto ad `orario`. Il metodo restituisce `true` in caso di successo, altrimenti lascia l'oggetto invariato e restituisce `false`. Tra i casi di fallimento vi sono anche quelli in cui un farmaco risulti già assunto oppure nessun farmaco debba essere assunto ad `orario`.

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

✓ `CasaDiCura cc2(cc);`

Costruttore di copia.

✓ `~CasaDiCura();`

Qualora sia necessario, implementare il distruttore.

✓ `cc.rimuoviFarmaco(idOspite, nome);`

Metodo che restituisce una nuova `CasaDiCura` uguale a `cc`, in cui tuttavia risulta rimosso il farmaco caratterizzato da `nome` ed associato all'ospite con `idOspite`. In caso di input invalidi o di elementi non trovati, il metodo restituisce una copia esatta di `cc`.

✓ `!cc;`

Operatore di negazione su `cc`, che modifica `cc` invertendo, per tutti i pazienti, lo stato di assunzione di tutti i farmaci aventi un orario di assunzione pari o antecedente le ore 12. Il metodo restituisce un riferimento al `cc` modificato.

Mediante il linguaggio C++, realizzare il tipo di dato astratto `CasaDiCura`, 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.**

Note per la consegna

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In tale caso, i docenti procederanno alla valutazione dell'elaborato **solo se** lo studente avrà completato l'autocorrezione. 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 valutato da parte dei docenti**.

Uscita che deve produrre il programma

--- PRIMA PARTE ---

Test costruttore e aggiungiFarmaco

Casa di Cura 'Giardino felice'

- Ospite 1:

1. 'Fluifort'

Non assunto

Ore 10

- Ospite 2:

1. 'Lasonil'

Non assunto

Ore 7

2. 'Aulin'

Non assunto

Ore 11

3. 'Tachipirina'

Non assunto

Ore 18

Test assumiFarmaci

Casa di Cura 'Giardino felice'

- Ospite 1:

1. 'Fluifort'

Non assunto

Ore 10

- Ospite 2:

1. 'Lasonil'

Assunto

Ore 7

2. 'Aulin'

Non assunto

Ore 11

3. 'Tachipirina'

Assunto

Ore 18

--- SECONDA PARTE ---

Test costruttore di copia

Casa di Cura 'Giardino felice'

- Ospite 1:

1. 'Fluifort'

Non assunto

Ore 10

- Ospite 2:

1. 'Lasonil'

Assunto

Ore 7

2. 'Aulin'

Non assunto

Ore 11

3. 'Tachipirina'

Assunto

Ore 18

Test eventuale distruttore

Distruttore chiamato

Test rimuoviFarmaco

Casa di Cura 'Giardino felice'

- Ospite 1:

1. 'Fluifort'

Non assunto

Ore 10

- Ospite 2:

1. 'Lasonil'

Assunto

Ore 7

2. 'Aulin'

Non assunto

Ore 11

Test operatore negazione

Casa di Cura 'Giardino felice'

- Ospite 1:

1. 'Fluifort'

Assunto

Ore 10

- Ospite 2:

1. 'Lasonil'

Non assunto

Ore 7

2. 'Aulin'

Assunto

Ore 11

3. 'Tachipirina'

Assunto

Ore 18