

Un `AmbienteVirtuale` contiene oggetti tridimensionali. Ogni oggetto ha un nome, una posizione all'interno del mondo virtuale individuata dalle coordinate reali x , y e z , e un identificatore univoco espresso come numero intero positivo. Realizzare le seguenti operazioni che possono essere effettuate in un ambiente virtuale:

- `AmbienteVirtuale a(n)`
Costruttore, che crea un ambiente virtuale a in grado di contenere al più n oggetti tridimensionali. Inizialmente, l'ambiente virtuale non contiene oggetti.
- `a1 = a`
Operatore di assegnamento, che sostituisce il valore dell'ambiente virtuale $a1$ con quello dell'ambiente virtuale a .
- `a.aggiungi(name, x, y, z)`
Operazione che aggiunge un oggetto nell'ambiente virtuale. L'oggetto ha nome $name$ e posizione (x, y, z) nel mondo virtuale. L'identificatore univoco dell'oggetto viene assegnato automaticamente dal sistema.
- `a.rimuovi(id)`
Operazione che rimuove l'oggetto con identificatore id , se presente, dall'ambiente virtuale. L'operazione restituisce `true` se la rimozione viene effettuata con successo; altrimenti, restituisce `false`.
- `!a`
Operatore di negazione logica, che restituisce il numero di oggetti contenuti nell'ambiente virtuale a .
- `~AmbienteVirtuale();`
Distruttore.
- `cout<<a;`
Operatore di uscita per il tipo `AmbienteVirtuale`. L'uscita visualizza gli oggetti posizionati nel mondo virtuale. Per ogni oggetto viene visualizzato il nome, l'identificatore e le coordinate. L'output seguente corrisponde a un mondo virtuale che contiene 2 oggetti, un oggetto di nome `sole`, id uguale a 3 e coordinate $(5, 1, 1)$, e un oggetto di nome `luna`, id uguale a 1 e coordinate $(1, 2, 3)$:

```
sole 3:(5,1,1)
luna 1:(1,2,3)
```

Mediante il linguaggio C++, realizzare il tipo di dato astratto `AmbienteVirtuale`, definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esaInf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- **Per una corretta stampa dell'elaborato**, non scrivere righe di codice di lunghezza eccessiva (mantenersi entro i margini imposti dall'ambiente *Dev-C++* cioè entro la linea verticale presente alla destra della pagina);
- **Per la Consegna:**
 - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

Condizione necessaria per la correzione dell'elaborato è che le prime tre funzioni siano state implementate correttamente.