

Il tipo di dato `Sistema` mantiene lo stato delle risorse gestite in un Centro di servizi informatici. Una risorsa è identificata in modo univoco da un nome formato da una sequenza di caratteri. Una risorsa può essere libera o occupata. Considerare le seguenti operazioni definite sul tipo di dato `Sistema`:

- `Sistema s;`
Costruttore di default che inizializza un sistema `s` con zero risorse.
- `s.aggiungi(nome);`
Operazione che aggiunge una nuova risorsa identificata da `nome` al sistema `s`, solo se il sistema non contiene già una risorsa con lo stesso `nome`. La risorsa aggiunta è libera. L'operazione restituisce `true` se l'inserimento ha successo; `false` altrimenti.
- `s1 = s2;`
Operatore di assegnamento, che sostituisce le risorse contenute nel sistema `s1` con quelle contenute nel sistema `s2`.
- `s.occupa(n);`
Operazione che inizializza lo stato della risorsa di nome `n` del sistema `s` ad occupata. L'operazione restituisce `true` se la risorsa è presente; `false` altrimenti.
- `!s;`
Operazione di negazione logica che restituisce `true` se il numero delle risorse libere è maggiore del numero delle risorse occupate; `false` altrimenti.
- `s1 += s2;`
Operazione di somma e assegnamento che aggiunge ad `s1` tutte le risorse libere di `s2` che non sono già presenti in `s1`.
- `cout<<s;`
L'operatore di uscita per il tipo `Sistema`.
L'operatore visualizza su linee differenti
 - il numero di risorse libere del sistema fra parentesi angolate, seguito dai nomi delle risorse libere separati dal carattere '-'
 - il numero di risorse occupate del sistema fra parentesi angolate, seguito dai nomi delle risorse occupate separati dal carattere '-'Per esempio, per un sistema che gestisce 5 risorse di cui tre libere e due occupate, l'uscita sarà:
 <3> lpnord - lpsud - disk1
 <2> lpovest - disk2
- `~Sistema();`
Distruttore.

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