

Un `Sistema` è formato da risorse, in numero limitato (la *capacità* del sistema). Ciascuna risorsa ha un nome, che consiste in una stringa di caratteri di lunghezza inferiore a 20. Una risorsa può essere libera od occupata.

Implementare le seguenti operazioni che possono essere effettuate su un `Sistema`:

- `Sistema s(M)`;  
Costruttore che inizializza un sistema `s` avente capacità pari a `M` risorse. Inizialmente, il sistema non contiene risorse.
- `Sistema s1(s)`;  
Costruttore di copia, che inizializza un sistema `s1` col valore del sistema `s`.
- `s.aggiungi(c)`  
Operazione che aggiunge una risorsa di nome `c` al sistema `s`. Tale risorsa è libera. L'operazione fallisce se il sistema già contiene un numero di risorse pari alla propria capacità. L'operazione fallisce inoltre se il sistema già contiene una risorsa di nome `c`. L'operazione restituisce *true* in caso di successo e *false* in caso di fallimento.
- `s.occupa(c)`  
Operazione che occupa la risorsa di nome `c` del sistema `s`. L'operazione fallisce se nel sistema `s` la risorsa `c` non è presente oppure non è libera. L'operazione restituisce *true* in caso di successo e *false* in caso di fallimento.
- `s.contiene(c)`  
Operazione che verifica se il sistema `s` contiene una risorsa di nome `c`. Se la verifica ha successo, l'operazione restituisce *true*, altrimenti l'operazione restituisce *false*.
- `s.elimina(c)`  
Operazione che elimina la risorsa libera di nome `c` dal sistema `s`. L'operazione fallisce se il sistema non contiene una risorsa di nome `c`, o se tale risorsa è occupata. L'operazione restituisce *true* in caso di successo e *false* in caso di fallimento.
- `~Sistema()`  
Distruttore.
- `cout << s`  
L'operatore di uscita per il tipo `Sistema`. L'uscita visualizza la capacità del sistema e il nome di ogni risorsa del sistema. Il seguente esempio mostra l'uscita per un sistema con capacità 4, in cui sono presenti due risorse (R2, R3) occupate, e una risorsa libera (R1):  
Capacita' del sistema: 4  
Risorse presenti nel sistema: 3  
Elenco risorse libere: R1  
Elenco risorse occupate: R2 R3

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti `Sistema`, definito dalle precedenti specifiche. Individuare le situazioni di errore e metterne in opera un corretto trattamento.

---

## NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- PER SVOLGERE L'ELABORATO,
- 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 LO SVOLGIMENTO DELL'ELABORATO, selezionare la voce **Identifica studente** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e inserire i dati richiesti;
- Per svolgere l'elaborato, aprire il progetto *esainf.dev* presente nel directory *c:\esame\esaInf* , e scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*, già presente nel progetto;
- Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. **NON MODIFICARE IL FILE.**
- Per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina);
- SE L'ELABORATO È STATO COMPILATO SENZA ERRORI, PRIMA DELLA CONSEGNA, selezionare la voce **Consegna** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e premere il tasto INVIO fino a quando non sparisce la finestra che è stata attivata.
- ASPETTARE AL PROPRIO POSTO DI ESSERE CHIAMATI PER VERIFICARE LA STAMPA DEL PROPRIO ELABORATO, FIRMARLO E CONSEGNARLO DEFINITIVAMENTE.