

Su un Biliardo sono disposte sfere, ogni sfera è identificata da un numero naturale e le sfere disposte su uno stesso biliardo hanno numeri *diversi*. Le operazioni che possono essere effettuate su un biliardo sono le seguenti:

✓ **inizializzaBiliardo(b, i);** [2pt]

Funzione che inizializza il biliardo b. Inizialmente, su tale biliardo è disposta una sola sfera, e tale sfera ha numero i.

✓ **aggiungiSfera(b, i);** [4pt]

Funzione booleana che aggiunge una nuova sfera con il numero i al biliardo b. Se esiste già una sfera con il numero i disposta sul biliardo, l'inserimento fallisce e la funzione restituisce false, altrimenti la funzione restituisce true. (Le sfere disposte su uno stesso biliardo hanno numeri *diversi*).

✓ **sottraiBiliardo(b, b1);** [7pt]

Funzione che elimina tutte le sfere del biliardo b1 dal biliardo b. La funzione restituisce quante sfere sono state effettivamente eliminate dal biliardo b.

Esempio:

Il biliardo b ha le sfere <2, 3, 7>.

Il biliardo b1 ha le sfere <2, 9>.

Dopo aver chiamato la `sottraiBiliardo(b, b1)`, b avrà le sfere <3, 7>.

✓ **vettoreSfere(v, n);** [6pt]

Funzione che dato un vettore di interi v, crea e restituisce un nuovo biliardo che contiene tutte le sfere il cui numero è presente nel vettore v. Il vettore v e la dimensione n del vettore sono passati come argomento alla funzione. (Le sfere disposte su uno stesso biliardo hanno numeri *diversi*).

Esempio:

Il vettore v è pari a [3, 8, 1, 2, 3] (dimensione 5).

Il biliardo creato da `vettoreSfere(v, 5)` avrà le sfere <1, 2, 3, 8>.

✓ **stampaBiliardo(b, "output.txt");** [6pt]

Funzione che stampa sul file di nome "output.txt" il biliardo b. L'uscita consiste nei numeri delle sfere disposte sul biliardo, in *ordine crescente*, separati da virgole e racchiusi tra parentesi angolate.

Esempio:

<1, 3, 5, 6>

Mediante il linguaggio C++, implementare il tipo Biliardo definito dalle precedenti specifiche utilizzando le **strutture**. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Esempio di funzione main()

```
int main(){
    Biliardo b;
    inizializzaBiliardo(b, 7);
    aggiungiSfera(b, 3);
    aggiungiSfera(b, 2);
    aggiungiSfera(b, 3);           // inserimento fallito
    stampaBiliardo(b, "output1.txt"); // File output1.txt
                                    // <2, 3, 7>

    Biliardo b1;
    inizializzaBiliardo(b1, 9);
    aggiungiSfera(b1, 2);         // b:<2,3,7>, b1:<2,9>
    cout << sottraiBiliardo(b, b1)<<endl; // stampa a video: 1

    stampaBiliardo(b, "output2.txt"); // File output2.txt
                                    // <3, 7>

    int v[5] = {3, 8, 1, 2, 3};
    Biliardo* p = vettoreSfere(v, 5);
    stampaBiliardo((*p), "output3.txt"); // File output3.txt
                                    // <1, 2, 3, 8>

    return 0;
}
```

Domande

1. Dato il numero naturale 145 in base 8, trovare la sua rappresentazione in base 16. [2pt]
2. Dire qual è l'uscita del seguente programma. [2pt]
(supporre che la variabile a abbia indirizzo 0x1A4E)

```
int main () {
    int a;
    int &b = a;
    int *p = &a;
    a = 10;
    cout << p << endl;
    cout << *p << endl;
    b++;
    cout << a << endl;
    return 0;
}
```

3. Errori sintattici ed errori semantici. Fare un esempio del primo e del secondo tipo di errore nei programmi. [1pt]