

## Esercizio – classe `StringSet`

Definire una classe di nome `StringSet` che realizza il tipo di dato astratto *insieme di stringhe*. La classe fornisce le seguenti funzioni proprie *public*:

- funzione `void add(x)`: aggiunge all'insieme un elemento `x`; se `x` è già presente, l'insieme rimane inalterato; ; nel caso non ci sia spazio sufficiente solleva un'eccezione "spazio non sufficiente"
- funzione booleana `contains(x)`: verifica se `x` appartiene o no all'insieme;
- funzione intera `size()`: restituisce la cardinalità dell'insieme;
- funzione booleana `S1 < S2`, con `S1` e `S2` di tipo `StringSet`: verifica se `S1` è un sottoinsieme ( $\subseteq$ ) di `S2`;
- funzione `S1 + S2`, con `S1` e `S2` di tipo `StringSet`: restituisce come suo risultato il nuovo insieme ottenuto dall'unione di `S1` e `S2` (gli eventuali elementi duplicati in `S1` e `S2` appaiono una volta soltanto nell'insieme risultato).

La classe fornisce anche due costruttori

- costruttore senza parametri: crea un insieme vuoto;
- costruttore con due parametri, `v` ed `n`, dove `v` è un array di elementi di tipo `string` ed `n` è un intero che indica il numero di elementi in `v`: crea un insieme contenente gli `n` elementi di `v` come suoi elementi.

E' inoltre prevista una funzione che ridefinisce l'operatore `<<`: `f_out << S` stampa tutti gli elementi dell'insieme di stringhe `S` sullo stream `f_out`, nel formato `{e1,e2,...,en}`.

Requisiti di implementazione. Realizzare l'insieme tramite un array statico di 100 elementi di tipo `string`. I parametri di tipo `StringSet` nelle funzioni `operator<` e `operator+` devono essere dichiarati (obbligatoriamente) const &.

Main di prova. La classe realizzata deve essere inserita nel main di prova messo a disposizione su 'elly' insieme a questo testo (file `stringset_main.cpp`). L'esecuzione del main, completato con la classe `StringSet`, deve dare i risultati riportati di seguito al main stesso. Non è ammesso apportare modifiche al main.

N.B. E' vietato: usare dichiarazioni friend, usare goto, usare variabili globali non motivate, definire `public` tutti i campi di una classe, usare ereditarietà a sproposito.