

Un `GestoreApp` rappresenta la lista di app in esecuzione in un certo momento su un dispositivo mobile, per esempio uno smartphone o un tablet. Ogni app è identificata da un nome di massimo 20 caratteri. Non esistono due app con lo stesso nome e i nomi delle app sono case-sensitive (cioè, “WhatsApp” è diverso da “WHATSAPP”). Una app entra nella lista quando viene *avviata*, e ne esce quando viene *chiusa*. La prima app della lista è quella in *foreground*, ovvero è quella visualizzata sullo schermo del dispositivo mobile. Tutte le altre sono in *background*. Implementare le seguenti operazioni che possono essere effettuate su un `GestoreApp`:

--- **PRIMA PARTE** --- (qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova sarà considerata insufficiente e pertanto non sarà corretta)

✓ `GestoreApp g;`

Costruttore di default che inizializza un `GestoreApp` vuoto, cioè senza nessuna app in esecuzione.

✓ `g += str;`

Operatore di somma e assegnamento che avvia una nuova app di nome `str` e la posiziona in foreground (cioè, come prima della lista). L'app che era precedentemente in foreground diventa seconda della lista, quella che era seconda diventa terza, e così via. Se una app con lo stesso nome era già in esecuzione, la lista resta inalterata.

✓ `cout << g;`

Operatore di uscita per il tipo `GestoreApp`. L'operatore stampa tra parentesi quadre la app in foreground, seguita dalle eventuali app in background separate da virgola e spazio, secondo questo formato:

```
[WhatsApp], Shazam, TripAdvisor, PlayStore
```

Se nessuna app è in esecuzione, stampa:

```
[]
```

--- **SECONDA PARTE** ---

✓ `g.foreground(str);`

Funzione che porta in foreground la app di nome `str`. La lista resta inalterata se una app con quel nome è già in foreground oppure non è in esecuzione.

✓ `g -= str;`

Operatore di sottrazione e assegnamento che chiude la app di nome `str`. Se la app da chiudere è in foreground, viene mandata in foreground la seconda della lista, se esiste. Se una app con quel nome non è in esecuzione, la lista resta inalterata.

✓ `g.chiudi_tutte();`

Funzione che chiude tutte le app in esecuzione.

✓ `~GestoreApp();`

Distruttore.

Mediante il linguaggio C++, realizzare il tipo di dato astratto `GestoreApp`, definito dalle precedenti specifiche. **Gestire le eventuali situazioni di errore.**

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore:

[]

Test operatore +=:

[WhatsApp]

[Shazam], TripAdvisor, PlayStore, WhatsApp

--- SECONDA PARTE ---

Test di foreground:

[TripAdvisor], Shazam, PlayStore, WhatsApp

[TripAdvisor], Shazam, PlayStore, WhatsApp

Test operatore -=:

[TripAdvisor], PlayStore, WhatsApp

[TripAdvisor], PlayStore, WhatsApp

Test di chiudi_tutte:

[]

Test distruttore:

[Facebook], YouTube

(g e' stato distrutto)