

In questo compito, deve essere implementata un'applicazione per gestire un calendario di eventi tramite una classe chiamata `Calendario`. Un evento è caratterizzato da un titolo (stringa di massimo 50 caratteri), una descrizione (stringa di massimo 100 caratteri), dal giorno (valore intero), dal mese (valore intero) e dall'anno (valore intero). **Tutte le stringhe** debbono essere valide: debbono avere lunghezza strettamente maggiore di zero e non possono contenere unicamente spazi.

--- Metodi invocati nella PRIMA PARTE di `main.cpp`: ---

✓ `Calendario c(titolo_calendario);`

Costruttore che crea un `Calendario` con titolo `titolo_calendario`. Se il titolo fornito non è valido il calendario deve intitolarsi "Senza nome". **Il titolo del calendario non ha limiti di caratteri.**

✓ `c.aggiungiEvento(titolo_evento, descrizione, giorno, mese, anno);`
 Aggiunge un nuovo evento con il titolo, la descrizione e la data specificati. Si suppone che non esistano anni bisestili (i.e. Febbraio ha sempre 28 giorni). **In una data specifica, può esistere al più un evento.** In caso di errore il metodo ritorna `false` (e l'evento non viene aggiunto), altrimenti l'evento viene aggiunto e il metodo ritorna `true`. **VERIFICARE ACCURATAMENTE CHE LA DATA SIA VALIDA**, altrimenti la funzione dovrà restituire `false` e non inserire l'evento.

✓ `cout << c;`

Stampa a video il contenuto del calendario `c`, ordinato per data crescente degli eventi. Deve essere mostrata anche la data dell'ultimo evento presente nel calendario (ossia di quello avente data più recente). Nel caso in cui non ci sia nessun evento, deve stampare:

```
Data ultimo evento: (vuota)
```

La cornice di "=" è composta da 30 caratteri "=". Di seguito viene mostrato un esempio di stampa a video di un calendario vuoto, dal titolo "Mio calendario":

```
=====
Calendario: Mio calendario
Data ultimo evento: (vuota)
=====
```

Altro esempio di calendario, avente titolo "Lavoro", ed avente due eventi:

```
=====
Calendario: Lavoro
Ultimo evento disponibile in data: 22/5/2025
%%%
Data: 18/4/2025
Titolo: Revisione
Descrizione: Meeting per correzione bozze
+++
Data: 22/5/2025
Titolo: Consegna
Descrizione: Consegna progetto
=====
```

Si noti come ogni evento successivo inizia con la stringa "+++", tranne il primo, che inizia con "%%%".

✓ `c.rimuoviEvento(giorno, mese, anno);`

Rimuove l'evento che si svolge alla data specificata. Se viene rimosso un evento, ritornare true, altrimenti ritornare false.

--- Metodi invocati nella SECONDA PARTE di main.cpp: ---

✓ `~Calendario();`

Implementare, se necessario, il distruttore, che libera la memoria associata agli eventi memorizzati nel calendario.

✓ `Calendario c2(c1);`

Costruttore di copia che inizializza un nuovo calendario c2 con gli stessi eventi e lo stesso titolo del calendario c1.

✓ `c.rimuoviEventiSuccessiviA(giorno, mese, anno);`

Rimuove tutti gli eventi che si svolgono dopo la data specificata (non inclusa), ritorna il numero di eventi eliminati.

✓ `c.cercaEventi(parolaChiave);`

Restituisce un nuovo calendario contenente tutti gli eventi che includono la parola chiave specificata nel titolo **oppure** nella descrizione. Il nuovo calendario ha come titolo `parolaChiave`. Suggerimento: può essere utilizzata la funzione `strstr` della libreria `cstring`, che ha i seguenti argomenti:

```
char* strstr(const char* string1, const char* string2);
```

La funzione `strstr()` trova la prima occorrenza di `string2` in `string1`. La funzione ignora il carattere `'\0'` di terminazione della cstringa `string2`, nel processo di ricerca della prima eventuale corrispondenza. La funzione `strstr()` restituisce un puntatore all'inizio della prima occorrenza di `string2` in `string1`. Se `string2` non viene trovata in `string1`, la funzione `strstr()` restituisce `nullptr`. Se `string2` punta a una stringa con lunghezza zero, la funzione `strstr()` restituisce `string1`.

Mediante il linguaggio C++, realizzare il tipo di dato astratto **Calendario**, definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc. **Gestire le eventuali situazioni di errore.**

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore e stampa a video:

```
=====
Calendario: Mio calendario
Data ultimo evento: (vuota)
=====
```

Test della aggiungiEvento (deve stampare 1101000):
1101000

Test stampa calendario:

```
=====
Calendario: Mio calendario
Data ultimo evento: 10/9/2024
%%
Data: 7/12/1989
Titolo: Compleanno
Descrizione: Compleanno Gojo
+++
Data: 8/9/2024
Titolo: Dentista
Descrizione: Appuntamento dentista
+++
Data: 10/9/2024
Titolo: Meeting
Descrizione: Discussione con team
=====
```

Test della rimuoviEvento (deve stampare 10):
10

```
=====
Calendario: Mio calendario
Data ultimo evento: 8/9/2024
%%
Data: 7/12/1989
Titolo: Compleanno
Descrizione: Compleanno Gojo
+++
Data: 8/9/2024
Titolo: Dentista
Descrizione: Appuntamento dentista
=====
```

Note per la consegna:

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In questo caso, i docenti procederanno alla valutazione dell'elaborato **solo** se lo studente avrà completato l'autocorrezione del proprio elaborato.

In **tutti** gli altri casi (per esempio, il programma non compila, non collega, non esegue o la prima parte dell'output non coincide con quella attesa), l'elaborato è considerato **insufficiente** e, pertanto, **non verrà corretto** dai docenti.

--- SECONDA PARTE ---

Test ulteriore della aggiungiEvento (deve stampare 11111):
11111

Test ricerca eventi che hanno keyword 'Meeting':

```
=====
Calendario: Meeting
Data ultimo evento: 18/4/2025
%%
Data: 10/1/2025
Titolo: Meeting
Descrizione: Meeting con il team.
+++
Data: 11/2/2025
Titolo: Meeting
Descrizione: Meeting con revisori
+++
Data: 18/4/2025
Titolo: Revisione
Descrizione: Meeting per correzione bozze
=====
```

Altro test rimuoviEvento:
Rimosso/i 1 evento/i
Calendario dopo la rimozione:

```
=====
Calendario: Meeting
Data ultimo evento: 11/2/2025
%%
Data: 10/1/2025
Titolo: Meeting
Descrizione: Meeting con il team.
+++
Data: 11/2/2025
Titolo: Meeting
Descrizione: Meeting con revisori
=====
```

Test costruttore di copia:

```
=====
Calendario: Eventi progetto
Data ultimo evento: 22/5/2025
%%
Data: 10/1/2025
Titolo: Meeting
Descrizione: Meeting con il team.
+++
Data: 10/2/2025
Titolo: Presentazione
Descrizione: Presentazione risultati
+++
Data: 11/2/2025
Titolo: Meeting
Descrizione: Meeting con revisori
+++
Data: 18/4/2025
Titolo: Revisione
Descrizione: Meeting per correzione bozze
+++
Data: 22/5/2025
Titolo: Consegna
Descrizione: Consegna progetto
=====
```