

Una `MediaPlaylist` rappresenta una lista di file multimediali. I file contenuti nella `MediaPlaylist` possono essere di tipo `AUDIO` o `VIDEO` e sono caratterizzati dal proprio titolo, ovvero da una stringa di al più `TITLEN=20` caratteri. Implementare le seguenti operazioni che possono essere effettuate su una `MediaPlaylist`:

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

✓ `MediaPlaylist mp;`

Costruttore di default che inizializza una `MediaPlaylist` vuota.

✓ `mp.inserisci(titolo, tipo);`

Operazione che aggiunge un nuovo file multimediale in testa alla playlist. Il file ha titolo `titolo` ed è di tipo `tipo`. Il file non viene aggiunto alla playlist se già presente. Se il nome del file è più lungo di `TITLEN`, il file viene aggiunto ugualmente e il suo nome viene troncato a `TITLEN`.

✓ `mp.elimina(titolo, tipo);`

Operazione che elimina il file avente titolo `titolo` e tipo `tipo` dalla `MediaPlaylist`. Se il file non è presente, la playlist rimane inalterata.

✓ `cout << mp;`

Operatore di uscita per il tipo `MediaPlaylist`. L'output è nel formato seguente:

```
[1]<We are the champions, Video>
[2]<Stairway to heaven, Audio>
```

Gli elementi sono numerati a partire da 1, secondo l'ordine della playlist. Il numero è racchiuso tra parentesi quadre. Titolo e tipo del file sono racchiusi tra parentesi angolari, separate da un carattere ``,`` e uno spazio. Nel caso in cui la playlist sia vuota, l'output mostrato è il seguente:

```
[-]
```

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

✓ `mp.riproduci(titolo, tipo);`

Operatore che manda in riproduzione il file avente nome `titolo`. L'operazione restituisce, nell'argomento `tipo`, il tipo del file riprodotto (audio o video). Qualora la playlist contenesse due file con lo stesso nome, viene riprodotto quello che compare prima nella playlist. L'operazione restituisce 1 se esiste un file con nome `titolo`, 0 altrimenti.

✓ `MediaPlaylist mp(vett, n);`

Costruttore che inizializza una `MediaPlaylist` a partire dagli elementi contenuti nel vettore `vett`. Il vettore ha dimensione `n`. Ogni elemento del vettore `vett` specifica il nome e il tipo di un file. I file appariranno nella playlist secondo lo stesso ordine secondo cui sono memorizzati nel vettore.

✓ `MediaPlaylist mp(mp2);`

Costruttore di copia per il tipo `MediaPlaylist`, che inizializza `mp` con il contenuto di `mp2`.

✓ `~MediaPlaylist();`

Distruttore.

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

---

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE---

Test del costruttore di default:

mp  
[-]

Test inserisci:

mp  
[1]<We are the champions, Video>  
[2]<Stairway to heaven, Audio>  
[3]<Another brick in the, Video>  
[4]<We are the champions, Audio>

Test elimina:

mp  
[1]<Stairway to heaven, Audio>  
[2]<Another brick in the, Video>  
[3]<We are the champions, Audio>

--- SECONDA PARTE---

Test costruttore MediaPlayerPlaylist(item\*, int)

mp2  
[1]<Michelle, Audio>  
[2]<Thriller, Video>

mp2  
[1]<Smoke on the water, Audio>  
[2]<Thriller, Video>

Test costruttore di copia:

mp3  
[1]<Smoke on the water, Audio>  
[2]<Thriller, Video>

mp3  
[1]<Sweet child o' mine, Video>  
[2]<Smoke on the water, Audio>  
[3]<Thriller, Video>

Test riproduci:

Riprodotta file VIDEO  
Riprodotta file AUDIO  
File non presente

Test del distruttore  
(mp4 distrutto)

---

**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**.