

L'esame di oggi ha a che fare con film e tessere cinematografiche.

Il tipo enumerazione **Genere** rappresenta le diverse tipologie di film: `per_bambini`, `vietato_ai_minori`, `impegnato`, `thriller`.

Gli elementi del tipo record **Film** rappresentano i singoli film, caratterizzati dal titolo, dal genere, e dalla durata (in minuti).

Il tipo **Stagione** è una lista semplice di **Film**.

Il tipo **Abbonamento** è un array di film, di lunghezza `N` fissata: al momento dell'acquisto dell'abbonamento, l'abbonato deve fissare gli `N` film della stagione per i quali ha diritto all'accesso gratuito.

```
typedef enum Genere {per_bambini, vietato_ai_minori,
impegnato, thriller} Genere;
typedef struct {char* titolo;
                Genere genere;
                int durata;}
                Film;
typedef struct Stagione {Film film;
                        struct Stagione *prossimo;}
                Stagione;

typedef Film Abbonamento[N];

typedef char* AbbList[N];
```

```
void Selezione(Stagione* s, Abbonamento abb, Genere g);
```

Questa procedura deve selezionare i primi `N` film di genere `g` presenti nella stagione `s` (che è una lista) ed assegnarli agli elementi della tabella `abb`. Se nella lista `s` sono presenti meno di `N` film di genere `g`, la procedura deve produrre un messaggio di errore sullo standard output..

- **void Stampa_a(Abbonamento a);**

Questa funzione deve produrre una rappresentazione completa sullo standard output dei film associati al valore di tipo `Abbonamento` passato come parametro attuale.

Genere Prevalente(Abbonamento abb, Stagione* s);

Questa funzione restituisce il genere che ha maggiori occorrenze tra i film dell'abbonamento `abb` relativo alla stagione `s`. In caso di parità, la funzione restituisce uno qualsiasi dei generi più presenti.

2.void Stampa_s(Stagione* s);

Questa funzione deve essere definita in modo ricorsivo, e deve produrre una rappresentazione completa sullo standard output dei film presenti nella lista passata come parametro attuale

ListaSpettacoli* filtra(ListaSpettacoli* s, Genere g);

Questa funzione restituisce una nuova lista di tipo `ListaSpettacoli`, che contiene solo gli spettacoli della lista `s` che non appartengono al genere `g`.

• **void stampa_piu_serate(Tessera t, ListaSpettacoli* lista);**

Questa funzione deve stampare il nome dello (o degli) spettacoli presenti nella tessera `t` che ha (o hanno, in caso di parità) il maggior numero di repliche. L'informazione relativa al numero di repliche per ogni spettacolo va cercata in `lista` (se lo spettacolo `t[i]` non è presente nella lista, il numero di repliche di quello spettacolo sarà ovviamente 0).