

Come ogni anno, nella ridente cittadina di Pasticci Marina, fervono i preparativi per l'importante e tanto attesa festa di Mezza Estate. Il comitato organizzatore è alle prese con la pianificazione degli eventi e la scelta delle attrattive che caratterizzano la festa, e ci ha chiesto di fornirgli un sistema che faciliti i loro compiti.

In particolare, la festa si svolgerà in più giornate, e per ogni giornata sono previste varie attrazioni di diverse categorie: eventi religiosi, spettacoli teatrali, eventi musicali, giochi, etc. Ogni attrazione ha un nome, una durata (per semplicità espressa in minuti), e un costo che dipende dalla categoria e dalla durata dell'attrazione. Inoltre, durante ogni giornata, le varie attrazioni si susseguono secondo un ordine che dipende dalla categoria di attrazione. Ad esempio, gli eventi religiosi dovrebbero svolgersi nella prima parte della giornata (quindi prima di tutti gli altri), mentre gli eventi musicali dovrebbero svolgersi nella parte finale (quindi dopo tutti gli altri).

Il sistema deve consentire di:

- aggiungere un'attrazione ad una giornata, in una posizione tale da rispettare le precedenze con le attrazioni già presenti;
- rimuovere un'attrazione da una giornata;
- verificare se una data attrazione è già stata inserita in una delle giornate;
- stampare i programmi delle varie giornate, con i dettagli di tutte le attrazioni previste.

Si chiede di progettare il sistema riportando la definizione delle classi necessarie alla sua realizzazione. Il sistema deve essere flessibile in caso di aggiunta di nuove attrazioni. Inoltre, deve essere riportata l'implementazione dei metodi mancanti della classe Festa illustrata nel riquadro e dei metodi richiesti sopra. Infine, discutere la complessità del metodo m1.

NOTA BENE:

Non saranno valutati elaborati con un numero maggiore di 3 classi, oltre alla classe Festa.

```
class Festa
{
public:
    Festa(unsigned int n): numGiornate(n)
        { festa = new Giornata[n]; }
    Festa (const Festa&);
    ~Festa();
    Festa& operator=(const Festa&);
    friend ostream& operator<<(ostream&,
        const Festa&);
    unsigned int calcolaCostoTotale() const;
    double m1() const
    {
        int m = numGiornate / 2;
        double costo = 0.0;

        for( int i = 0; i < m; i++ ){
            if( festa[ i ].getCosto() > costo )
                costo = festa[ i ].getCosto();
            else{
                for( int j = m + 1; j < numGiornate; j++ )
                    if( festa[ j ].getCosto() > costo )
                        costo = festa[ j ].getCosto();
            }
        }
        return costo;
    }
private:
    Giornata* festa;
    unsigned int numGiornate;
};
```