

Un `Treno` ha un nome ed è formato da vagoni. Il nome del treno consiste in una stringa di lunghezza massima 20 caratteri. I vagoni sono numerati a partire da 1. Ogni vagone ha un numero di posti uguale a 8, e i posti numerati a partire da 1. Si possono aggiungere e togliere vagoni dal treno. Ogni posto può essere libero o occupato. Implementare le seguenti operazioni che possono essere effettuate su un `Treno`:

- `Inizializza(t, name);`
Funzione che inizializza un treno `t`. Il treno ha nome `name` ed è formato da 1 vagone. Inizialmente tutti i posti del vagone sono liberi.
- `OccupatoPosto(t, v, i);`
Funzione che occupa il posto `i` del vagone `v` del treno `t`. Se l'operazione fallisce, la funzione restituisce `false`; altrimenti la funzione restituisce `true`.
- `LiberaPosto(t, v, j);`
Funzione che libera il posto `j` del vagone `v` del treno `t`. Se l'operazione fallisce, la funzione restituisce `false`; altrimenti la funzione restituisce `true`.
- `Occupato(t, k);`
Funzione che occupa `k` posti in un vagone del treno `t`. Il vagone è il primo con almeno `k` posti liberi. Se non ci sono vagoni con numero di posti liberi sufficienti, l'operazione fallisce e la funzione restituisce `false`; altrimenti la funzione restituisce `true`.
- `AggiungiVagoni(t, m);`
Funzione che aggiunge `m` vagoni in fondo al treno. Tutti i posti dei vagoni sono liberi.
- `Compatta(t);`
Funzione che riduce il numero di vagoni occupati del treno al minimo numero possibile. I posti vengono occupati a partire dal primo vagone, quando il vagone è pieno, si assegnano i posti del vagone successivo. I vagoni liberi sono solo quelli in fondo al treno.
- `Stampa(t);`
Funzione di uscita di un `Treno`, che stampa il nome del treno e per ogni vagone, i posti occupati. Prima viene stampato il nome del treno, poi su una nuova riga vengono stampati i vagoni. Per ogni vagone, i posti occupati sono racchiusi fra parentesi quadre e separati dal carattere `'`. La stampa avviene sempre mantenendo l'ordine dei vagoni.
Nel caso seguente il nome del treno è: `Intercity 657`. Il treno ha 5 vagoni, i posti occupati del primo vagone sono quelli di numero 2 e 6; il secondo, il quarto e il quinto vagone hanno tutti i posti liberi.

`Intercity 657`

`[2, 6] [] [1, 3, 4, 8] [] []`

Mediante il linguaggio C++, realizzare il tipo `Treno` definito dalle precedenti specifiche utilizzando le strutture. Gestire le eventuali situazioni di errore.

Esempio di funzione main()

Contenuto dello stream d'output standard mostrato nel commento */* */*

```
int main(){
    Treno t;
    Inizializza (t, "Intercity 657");
    Stampa(t);
    /*
Intercity 657
[] */

    OccupaPosto(t, 1, 6);
    OccupaPosto(t, 1, 2);
    OccupaPosto(t, 1, 3);
    /*
Intercity 657
[2, 3, 6] */

    LiberaPosto(t, 1, 3);
    Stampa(t);
    /*
Intercity 657
[2, 6] */

    AggiungiVagoni(t, 4);
    Stampa(t);
    /*
Intercity 657
[2, 6] [] [] [] */

    Occupa(t, 7);
    Stampa(t);
    /*
Intercity 657
[2, 6] [1, 2, 3, 4, 5, 6, 7] [] [] */

    LiberaPosti(t, 1, 2);
    Stampa(t);
    /*
Intercity 657
[6] [1, 2, 3, 4, 5, 6, 7] [] [] */

    CompattaTreno(t);
    Stampa(t);
    /*
Intercity 657
[1, 2, 3, 4, 5, 6, 7, 8] [] [] [] */

    return 0;
}
```