

Una `MappaNavale` mantiene informazioni sulla posizione di navi in un'area marina di dimensioni $M \times N$. Ogni nave è rappresentata da un identificatore, il quale è un numero intero positivo strettamente maggiore di zero. Le navi possono essere posizionate sia in maniera orizzontale che verticale. Non è possibile avere su una `MappaNavale` due navi con lo stesso identificatore, e non possibile avere sovrapposizioni di navi. Infine, le navi possono essere di dimensione variabile.

✓ **`inizializzaMappaNavale(A, M, N);`** [3pt]

Funzione che inizializza la `MappaNavale` `A` di dimensioni $M \times N$. Inizialmente non ci sono navi posizionate nella `MappaNavale`.

✓ **`aggiungiNave(A, id, dim, p, i, j);`** [5pt]

Funzione booleana che aggiunge una nave avente identificatore `id` e dimensione `dim` nella `MappaNavale` `A` a partire dalla posizione (i, j) . Il parametro `p` è un carattere che può assumere solo due valori: 'V' e 'O', che indicano rispettivamente il verso di posizionamento della nave (verticale/orizzontale). Se l'operazione di inserimento ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

✓ **`lunghezzaNave(A, id);`** [4pt]

Funzione che cerca la nave avente identificatore `id` nell'area `A` e restituisce la sua lunghezza. Se non esiste una nave avente tale identificatore, la funzione restituisce zero.

✓ **`stampaMappaNavale(A);`** [4pt]

Funzione che stampa a video la `MappaNavale`. L'uscita ha il seguente formato:

```
1 1 0 2 0
4 0 0 0 0
4 3 3 3 3
4 0 9 9 0
```

In questo caso la `MappaNavale` ha dimensioni 4×5 , e sono posizionate 5 navi: una nave avente dimensione 1 (`id:2`), due navi aventi dimensione 2 (`id:1` e `9`), una nave avente dimensione 3 (`id:4`), ed una nave avente dimensione 4 (`id:3`). Tutte le aree non occupate da navi sono rappresentate dal numero zero.

✓ **`eliminaNave(A, id);`** [4pt]

Funzione booleana che rimuove dalla `MappaNavale` `A` la nave avente identificatore `id`. Se l'eliminazione ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

✓ **`listaNavi(A);`** [5pt]

Funzione che restituisce una lista di elementi, uno per ogni nave presente nella `MappaNavale` `A`. Ogni elemento della lista deve contenere l'identificatore della nave. La lista non deve contenere duplicati e deve mantenere gli identificatori in ordine crescente.

Mediante il linguaggio C++, implementare il tipo `MappaNavale` definito dalle precedenti specifiche utilizzando le **strutture**. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Esempio di funzione main()

```

int main() {

    MappaNavale A;
    inizializzaMappaNavale(A, 4, 5);
    stampaMappaNavale(A);

    aggiungiNave(A, 3, 4, 'O', 2, 1);
    aggiungiNave(A, 4, 3, 'V', 1, 0);
    aggiungiNave(A, 5, 3, 'O', 1, 0);
    stampaMappaNavale(A);

    cout << lunghezzaNave(A, 4); // 3
    cout << lunghezzaNave(A, 5); // 0

    elem *p0 = listaNavi(A);

    eliminaNave(A, 3);
    stampaMappaNavale(A);

    return 0;
}

```

```

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

```

```

0 0 0 0 0
4 0 0 0 0
4 3 3 3 3
4 0 0 0 0

```



```

0 0 0 0 0
4 0 0 0 0
4 0 0 0 0
4 0 0 0 0

```

Domande

1. **[2pt]** Rappresentare il numero esadecimale A1B in base 8.
2. **[2pt]** Scrivere una funzione ricorsiva che prende in ingresso un array di interi di dimensione N e un numero k, e restituisce quanti elementi nell'array sono maggiori o uguali a k.
3. **[1pt]** Indicare a video l'output del seguente programma:

```

int main() {
    char v[10];
    cout << sizeof(v) << endl;
    cout << sizeof(v[7]) << endl;
    cout << sizeof(&v[0]) << endl;
}

```