

Un'Aula di personal computer è composta da 50 postazioni. Le postazioni sono numerate a partire da 1. Ad ogni istante ciascuna postazione può essere libera oppure occupata da un utente. Ciascun utente è identificato attraverso una stringa di al più 15 caratteri. Le funzioni che possono essere chiamate su un'Aula sono le seguenti:

✓ **inizializzaAula(A);** [4pt]

Funzione che inizializza l'Aula A. All'inizio tutte le postazioni sono libere.

✓ **aggiungiUtente(A, id);** [5pt]

Funzione che aggiunge l'utente avente identificatore `id` all'Aula A, in corrispondenza della prima postazione libera. Qualora l'aula sia piena, la funzione restituisce `false`. La funzione fallisce anche nel caso in cui si tenti di aggiungere un utente con `id` uguale a quello di un utente già presente in aula, nel qual caso restituisce ancora `false`. Infine, restituisce `true` quando l'inserimento ha successo.

✓ **liberaPostazione(A, i);** [5pt]

Funzione che libera la postazione `i`-esima dell'Aula A, qualora sia occupata da un utente. La funzione restituisce `true` se la postazione `i`-esima viene liberata, `false` altrimenti.

*Esempio:* se `i=1`, allora viene liberata la prima postazione.

✓ **stampaAula(A);** [5pt]

Funzione che stampa a video l'Aula A. La stampa ha il seguente formato:

AULA:

POSTAZIONE1:Bianchi

POSTAZIONE2:<libera>

POSTAZIONE3:Rossi

POSTAZIONE4:<libera>

.

.

POSTAZIONE50:Verdi

L'output mostrato corrisponde ad un'Aula di cui la prima, la terza e l'ultima postazione sono occupate rispettivamente da Bianchi, Rossi e Verdi. Tutte le altre postazioni sono libere.

✓ **listaUtenti(A);** [6pt]

Funzione che restituisce una *lista* che contiene gli utenti che al momento occupano l'Aula A.

La lista deve essere ordinata rispetto alla posizione nell'Aula, dalla più piccola alla più grande.

Se l'Aula A ha tutte le postazioni libere, la funzione restituisce una *lista vuota*.

*Esempio:* Nel caso in cui l'aula sia nella situazione della funzione di stampa, la lista restituita avrà, in ordine:

Bianchi → Rossi → Verdi

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

### Esempio di funzione main()

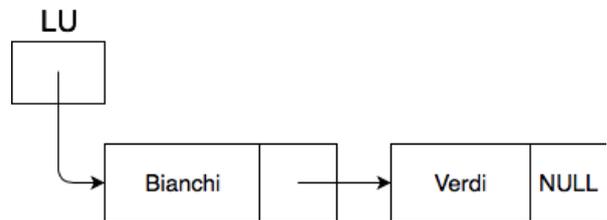
```
int main(){
    Aula A;
    inizializzaAula(A);
    aggiungiUtente(A, "Bianchi");
    aggiungiUtente(A, "Rossi");
    aggiungiUtente(A, "Rossi"); // non inserisce (duplicato)
    stampaAula(A); // POSTAZIONE1:Bianchi
                    // POSTAZIONE2:Rossi
                    // POSTAZIONE3:<libera>
                    // ... continua: tutte libere

    aggiungiUtente(A, "Verdi");
    liberaPostazione(A, 2); // Libera la postazione
                           // occupata da Rossi

    stampaAula(A); // POSTAZIONE1:Bianchi
                  // POSTAZIONE2:<libera>
                  // POSTAZIONE3:Verdi
                  // ... continua: tutte libere

    elem *LU = listaUtenti(A);

    return 0;
}
```



### Domande

1. **[2pt]** Rappresentare il numero intero -51 in complemento a 2 su 8 bit, se rappresentabile.
2. **[2pt]** Scrivere una funzione ricorsiva che prende in ingresso un array di interi di dimensione N, e restituisce il numero di elementi del vettore che sono pari.

Esempio: con l'array <-1,3,0,8> la funzione restituisce 2.

3. **[1pt]** Indicare l'uscita a video del seguente frammento di codice:

```
char c = 'p';
c = c + 5;
cout << c;
```