```cpp
#include <iostream>
#include <cstring>

using namespace std;


/* Il numero N è ininfluente, poteva essere 50,
 * come poteva essere 10000000.
 * Nella soluzione viene lasciato a 5 per comodita'
 */
const int N = 5;

struct postazione{
    char utente[16];
    bool occupata;
};

struct Aula{
    postazione v[N];
};

void inizializzaAula(Aula &A){
    for(int i=0;i<N;i++)
        A.v[i].occupata = false;
}

bool aggiungiUtente(Aula &A, const char* id){

    if(strlen(id)>15)
        return false;

    //controllo prima che l'utente non ci sia

    for(int i=0;i<N;i++)
        if(A.v[i].occupata && strcmp(A.v[i].utente, id)==0)
            return false;

    for(int i=0;i<N;i++) {
        if (A.v[i].occupata == false) {
            A.v[i].occupata = true;
            strcpy(A.v[i].utente, id);
            return true;
        }
    }
    //Se sono arrivato qui, sono uscito dal for e quindi non
ho aggiunto
    // ritorno false
    return false;
}

bool liberaPostazione(Aula &A, int i){

    // controllo che i non sia fuori range
    if(i<=0 || i>N)
        return false;

    // controllo se è libera
```

```
    // Attenzione! Si conta a partire da 1, ma negli array si
conta da 0,
    // quindi bisogna fare i-1, altrimenti si esce dall'array
    if(A.v[i-1].occupata == false)
        return false;

    // libero la postazione
    A.v[i-1].occupata = false;
    return true;
}

void stampaAula(Aula &A){

    cout << "AULA" << endl;
    for(int i=0;i<N;i++){
        cout << "POSTAZIONE" << (i+1) << ": ";
        if(A.v[i].occupata == true)
            cout << A.v[i].utente << endl;
        else
            cout << "<libera>" << endl;
    }
}

struct elem{
    char utente[16];
    elem *pun;
};

void inserisci_in_fondo(elem *&testa, const char* utente){

    elem *p, *q;
    for(q = testa; q!=NULL; q=q->pun)
        p = q;

    elem *r = new elem;
    strcpy(r->utente, utente);
    r->pun = NULL;

    if(testa == NULL)
        testa = r;
    else
        p->pun = r;
}

elem* listaUtenti(Aula &A){

    elem *testa = NULL;

    for(int i=0;i<N;i++){
        if(A.v[i].occupata == true)
            inserisci_in_fondo(testa, A.v[i].utente);
    }

    return testa;
}

void stampa_lista(elem *testa){
```

```
    cout  << "Lista Utenti: ";
    for(elem *q=testa;q!=NULL;q=q->pun) {
        cout << q->utente;
        if(q->pun != NULL)
            cout << "->";
    }
    cout << endl;
}

int f(int v[], int n){
    if(n==0)
        return 0;

    if(v[n-1]%2 == 0)
        return 1+f(v,n-1);
    else
        return 0+f(v,n-1);
}


int main() {
    Aula A;
    inizializzaAula(A);

    aggiungiUtente(A, "Bianchi");
    aggiungiUtente(A, "Rossi");
    aggiungiUtente(A, "Rossi");

    stampaAula(A);

    aggiungiUtente(A, "Verdi");
    liberaPostazione(A, 2);

    stampaAula(A);

    elem *LU = listaUtenti(A);

    stampa_lista(LU);

    // PROVA funzione ricorsiva
    int v[7] = {-1, 3, 0, 8, 15, 26, -30};

    int res = f(v, 7);
    cout << res;

    return 0;
}

/* RISPOSTE DOMANDE APERTE:
 *
 * Domanda 1: IL numero è rappresentabile.
 * La sua rappresentazione in complemento a 2 su 8bit
 * e' 11001101
 *
 * Domanda 2: vedere funzione f(int v[], int n) sopra
 *
 * Domanda 3: il codice stampa la lettera 'u'
```

```
 *
 */
```