

# Prova di Laboratorio del 9 Gennaio 2018

(Tempo a disposizione - 2h)

Dato lo scheletro di codice qui riportato, lo si completi aggiungendo l'implementazione della funzione e della procedura richieste.

bozza.c

```
#include <stdio.h>
#include <stdlib.h>

//List structure:
struct El {
    int info;
    struct El *next;
};

typedef struct El ElementoLista;
typedef ElementoLista *ListaDiElementi;

// Declarations of function and procedure to be implemented:

int readList(ListaDiElementi *lista);
void filterLists(ListaDiElementi *lista1, ListaDiElementi lista2
);

//Function that prints all the elements of the list:
void printList(ListaDiElementi list) {
    printf("(");
    while (list != NULL) {
        printf("%d ", list->info);
        list = list->next;
    }
    printf(")\n");
}

int main() {
    ListaDiElementi first_list = NULL, second_list=NULL;
    int size1=0,size2=0;

    //Read first list:
    size1 = readList(&first_list);

    //Read second list:
```

```

size2 = readList(&second_list);

//Print first list:
printf("Prima lista (%d elementi)\n", size1);
printList(first_list);

//Print second list:
printf("Seconda lista (%d elementi)\n", size2);
printList(second_list);

//Filter from first list the elements of second list and
write size:
filterLists(&first_list, second_list);

//Print filtered list:
printf("Lista filtrata\n");
printList(first_list);

return 0;
}

```

La funzione da implementare deve rispettare la seguente specifica:

- **readList**: Legge dallo standard input una sequenza di numeri interi *ordinati in maniera strettamente crescente* e termina automaticamente l'acquisizione alla prima occorrenza di un numero che non rispetta l'ordinamento (l'intero che viola l'ordinamento **non va inserito** nella lista). Gli interi devono essere memorizzati, nell'ordine di acquisizione, in una lista concatenata opportunamente allocata. La funzione **readList** restituisce un intero pari al numero di elementi presenti nella lista creata. Per esempio, supponendo che venga acquisita la sequenza (5, 8, 15, 9) la lista dovrebbe essere la seguente (e sarebbe la stessa se si acquisisse la sequenza (5, 8, 15, 15)), e la funzione restituirebbe il valore 3:

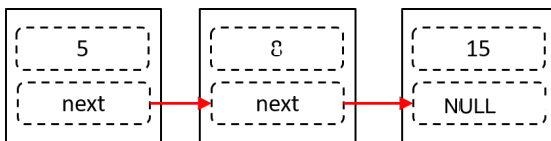


Figura 1: Esempio di acquisizione e memorizzazione di una lista.

La procedura da implementare deve rispettare la seguente specifica:

- **filterList**: date due liste *list1* e *list2* di lunghezza qualsiasi *ordinate* e *che non contengono duplicati*, la funzione elimina dalla lista *list1* tutti gli elementi che sono presenti anche nella lista *list2*. Ad esempio, date due liste rispettivamente contenenti gli interi (5, 8, 10, 15, 20, 24) e (5, 8, 9, 10, 20, 21) la lista *list1* risultante dovrà essere quella rappresentata in Figura 2.

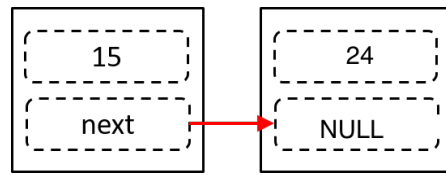


Figura 2: Esempio di prima lista filtrata.

## Esempio

### Input

5  
8  
15  
15  
5  
8  
9  
10  
20  
21  
3

### Output

Prima lista (3 elementi)  
(5 8 15 )  
Seconda lista (6 elementi)  
(5 8 9 10 20 21 )  
Lista filtrata  
(15 )