

## Definizione di lista (elementi interi)

```
struct list
{
    int el;
    struct list *next;
};
```

---

---

---

---

---

---

---

---

## Esercizio

Scrivere una funzione per cercare un numero  $x$  in una lista circolare di interi. La funzione deve restituire NULL se il numero non esiste.

Pre condizioni:

la funzione prende in ingresso un numero  $x$  intero e una lista  $L$  circolare a valori interi

Post condizioni:

se  $x \in L$ , la funzione restituisce (il puntatore al)l'elemento di  $L$  che codifica  $x$   
altrimenti restituisce NULL

---

---

---

---

---

---

---

---

## Svolgimento iterativo

```
struct list *cerca(struct list *l, int x)
{
    struct list *l2 = l;
    /* se la lista è vuota, restituisce NULL */
    if (l == NULL) return NULL;
    /* per ogni elemento della lista */
    do
    {
        if (l2->el == x) return l2;
        /* elemento successivo */
        l2 = l2->next;
    } while(l2 != l);
    return NULL;
}
```

---

---

---

---

---

---

---

---

## Esercizio

Scrivere una funzione ricorsiva che cerchi un elemento in una lista circolare. Utilizzare la tail recursion.

Pre condizioni:

la funzione prende in ingresso una lista circolare e un elemento

Post condizioni:

la funzione restituisce 1 se l'elemento è stato trovato, 0 altrimenti.

---

---

---

---

---

---

---

---

## Svolgimento

```
int ricerca(struct list *l, struct list *s, int x)
{
    if (l->next == s) return l->el == x;
    return l->el == x || ricerca(l->next, s, x);
}
```

```
int ricerca_tr(struct list *l, int x)
{
    if (l == NULL) return 0;
    return ricerca(l, l, x);
}
```

---

---

---

---

---

---

---

---