

Esercizio 1

- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa

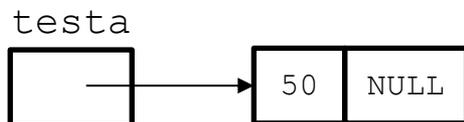
testa

NULL

```
struct elem{
    int info;
    elem *pun;
};
```

Esercizio 1

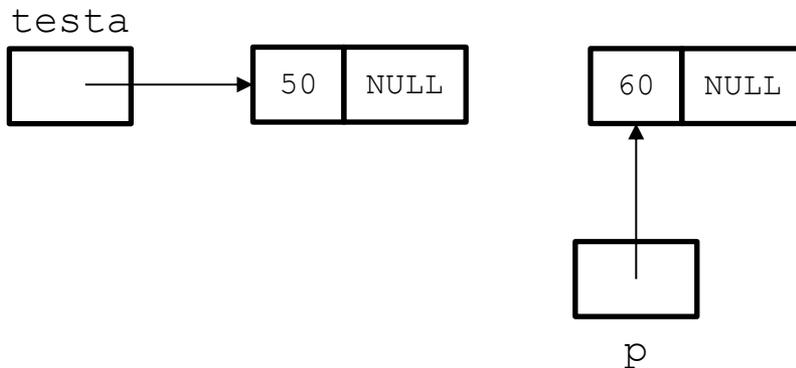
- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{
    int info;
    elem *pun;
};
```

Esercizio 1

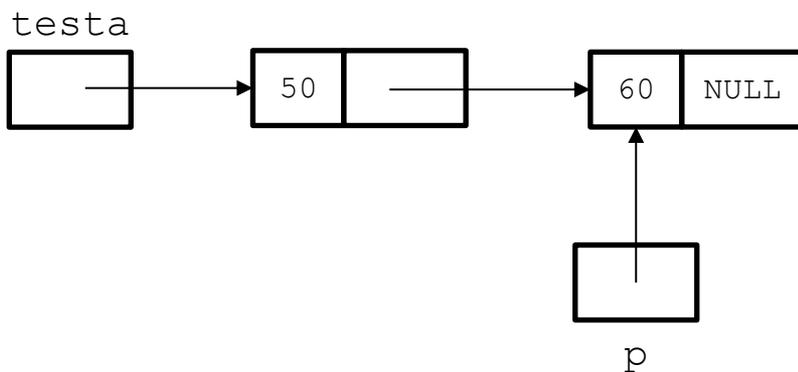
- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{  
    int info;  
    elem *pun;  
};
```

Esercizio 1

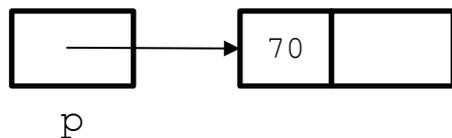
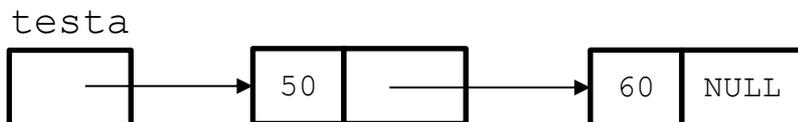
- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{  
    int info;  
    elem *pun;  
};
```

Esercizio 1

- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa

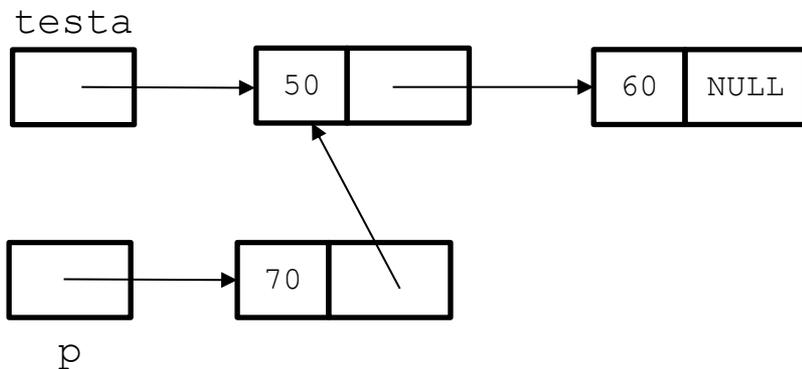


```

struct elem{
    int info;
    elem *pun;
};
  
```

Esercizio 1

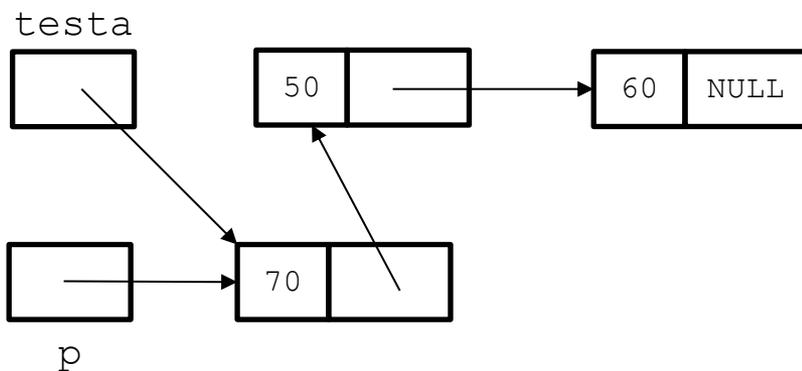
- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{
    int info;
    elem *pun;
};
```

Esercizio 1

- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{  
    int info;  
    elem *pun;  
};
```

Esercizio 1

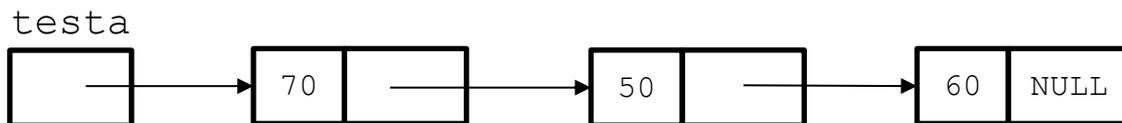
- Scrivere un main in cui si fanno le seguenti operazioni:
 1. Si crea una lista di interi vuota
 2. Si inserisce l'elemento intero 50
 3. Si inserisce l'elemento 60 in fondo
 4. Si inserisce un terzo elemento 70 in testa



```
struct elem{
    int info;
    elem *pun;
};
```

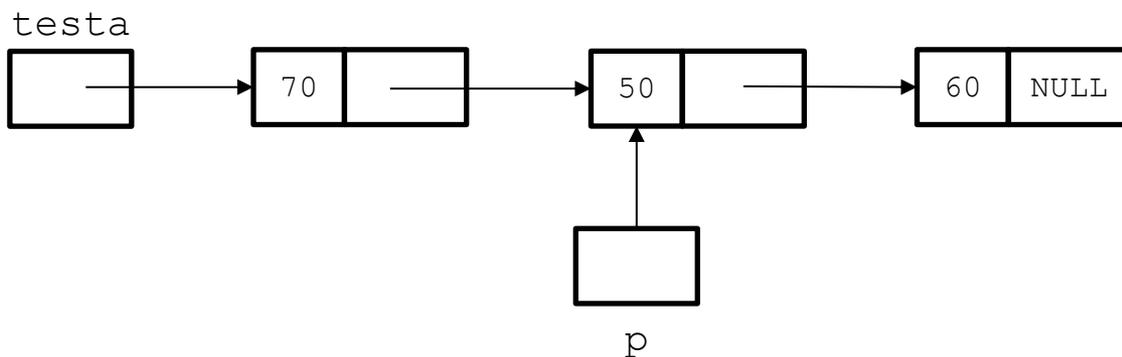
Esercizio 1 - Parte

- Scrivere inoltre una funzione 'stampa_lista' che prenda per argomento la lista appena creata e ne mostri a video il valore di tutti i suoi elementi
- Quindi con la lista qui in basso deve stampare 70 -> 50 -> 60 (o anche più semplicemente 70 50 60).



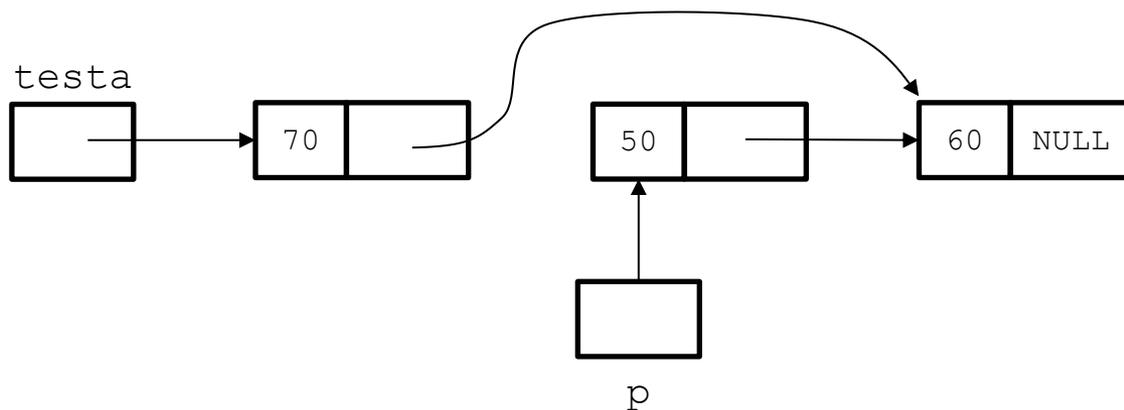
Esercizio 1

- Eliminare poi il secondo elemento (50) dalla lista e stampare di nuovo la lista.
Deve stampare 70 -> 60 (o più semplicemente 70 60).



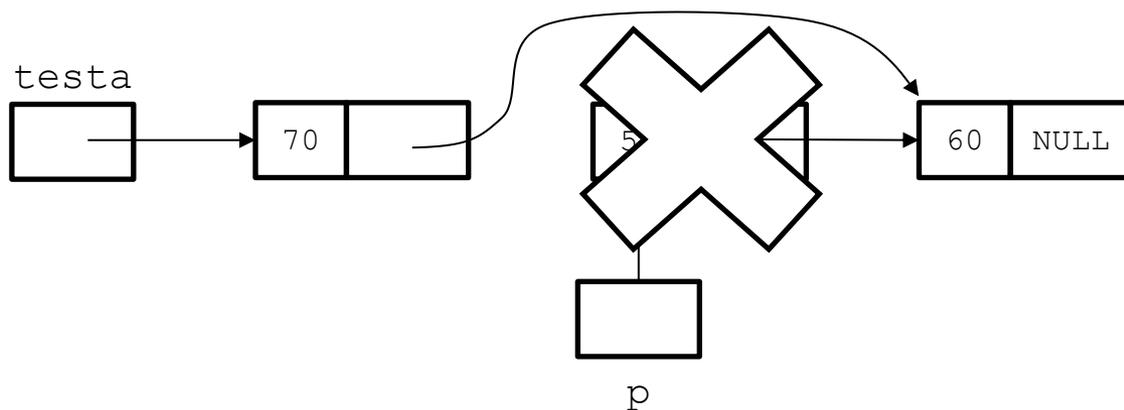
Esercizio 1

- Eliminare poi il secondo elemento (50) dalla lista e stampare di nuovo la lista.
Deve stampare 70 -> 60 (o più semplicemente 70 60).



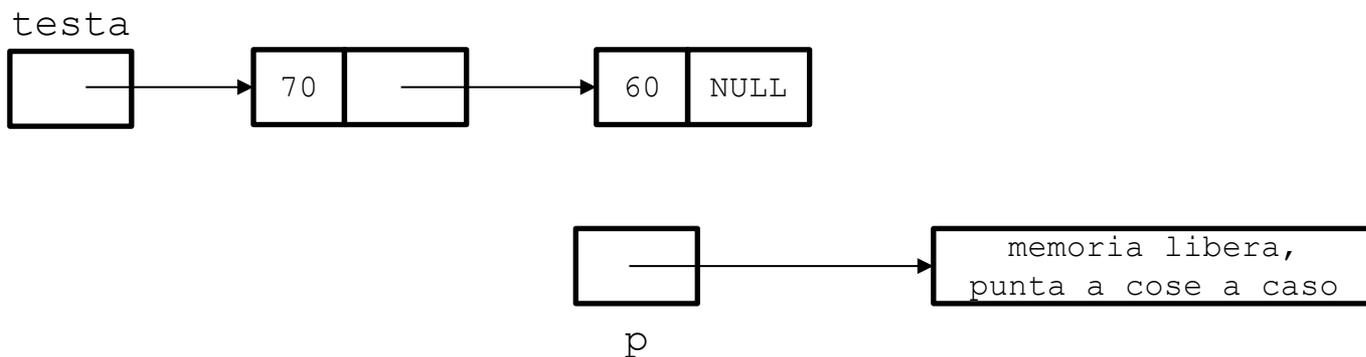
Esercizio 1

- Eliminare poi il secondo elemento (50) dalla lista e stampare di nuovo la lista.
Deve stampare 70 -> 60 (o più semplicemente 70 60).



Esercizio 1

- Eliminare poi il secondo elemento (50) dalla lista e stampare di nuovo la lista.
Deve stampare 70 -> 60 (o più semplicemente 70 60).



Esercizio 2

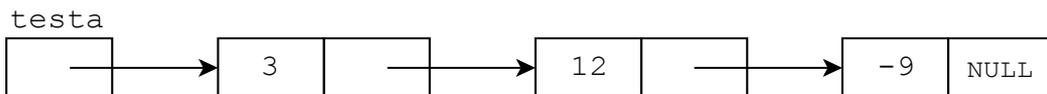
Sia data la struttura seguente:

```
struct elem {
    int info;
    elem* pun;
};
```

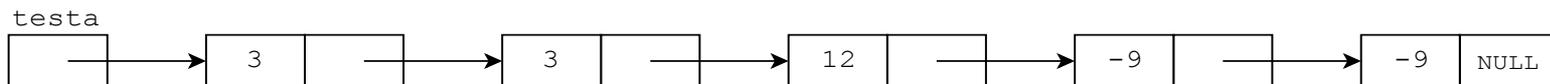
Scrivere una funzione che:

1. Prende in ingresso una lista di elementi di tipo elem ed un intero n.
2. Modifica la lista duplicando tutti gli elementi il cui campo informazione è minore di n, mantenendo l'elemento inserito in posizione adiacente all'elemento presente nella lista.

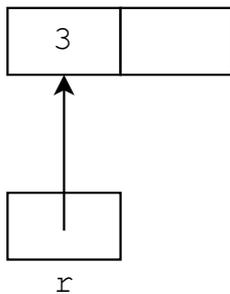
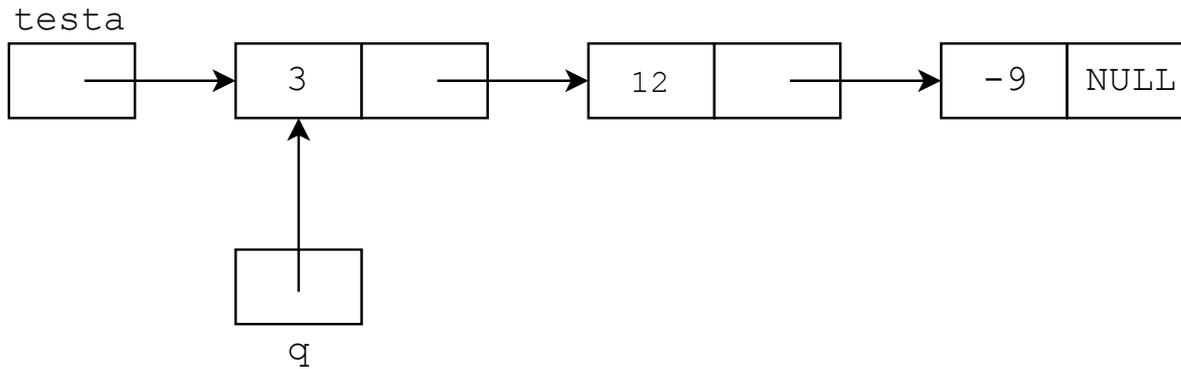
Per esempio, se la funzione viene chiamata con la lista seguente



ed n è uguale a 5, la lista modificata sarà la seguente:



Esercizio 2



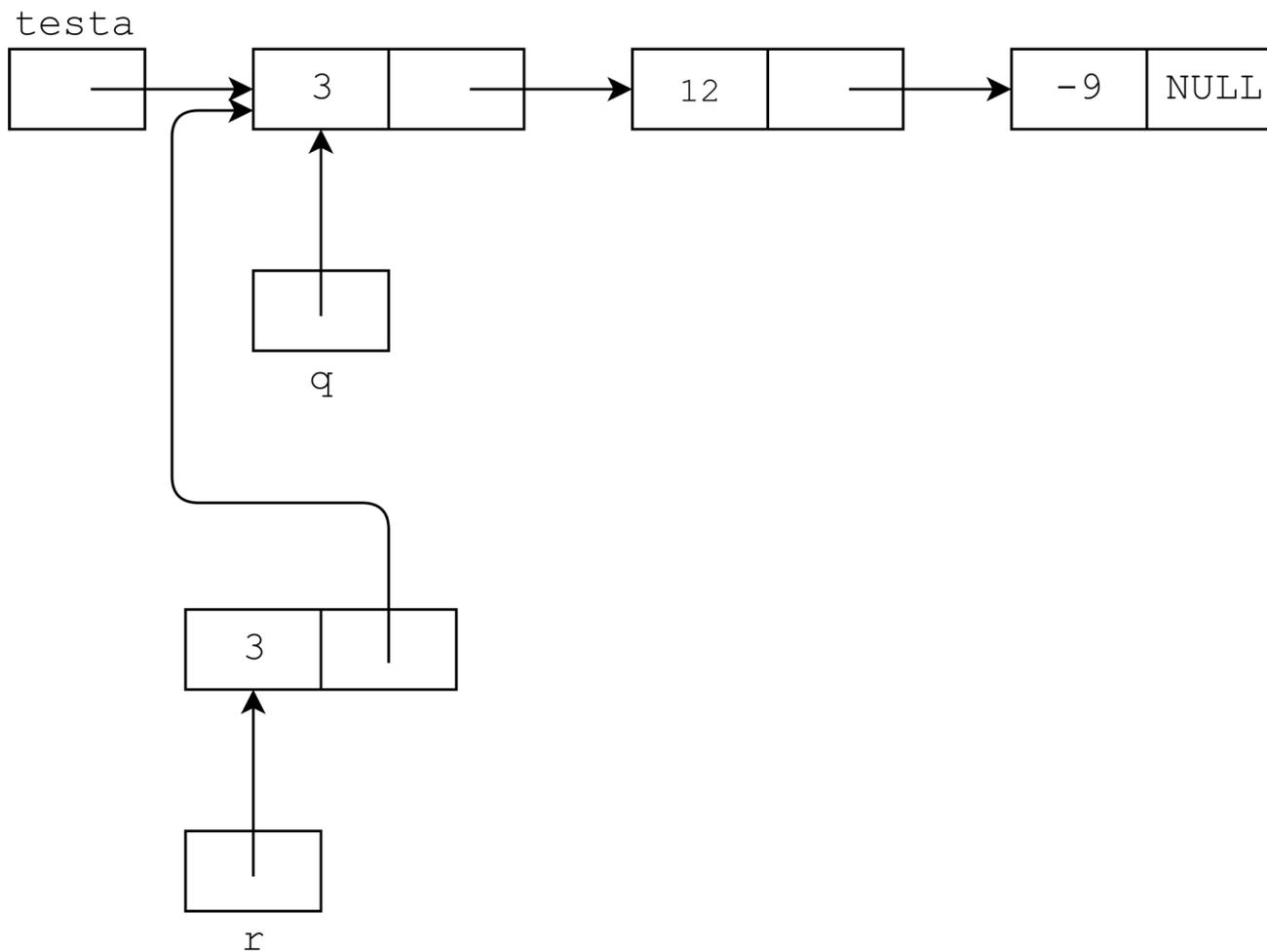
condizione cardine

$q \rightarrow \text{info} < 5?$

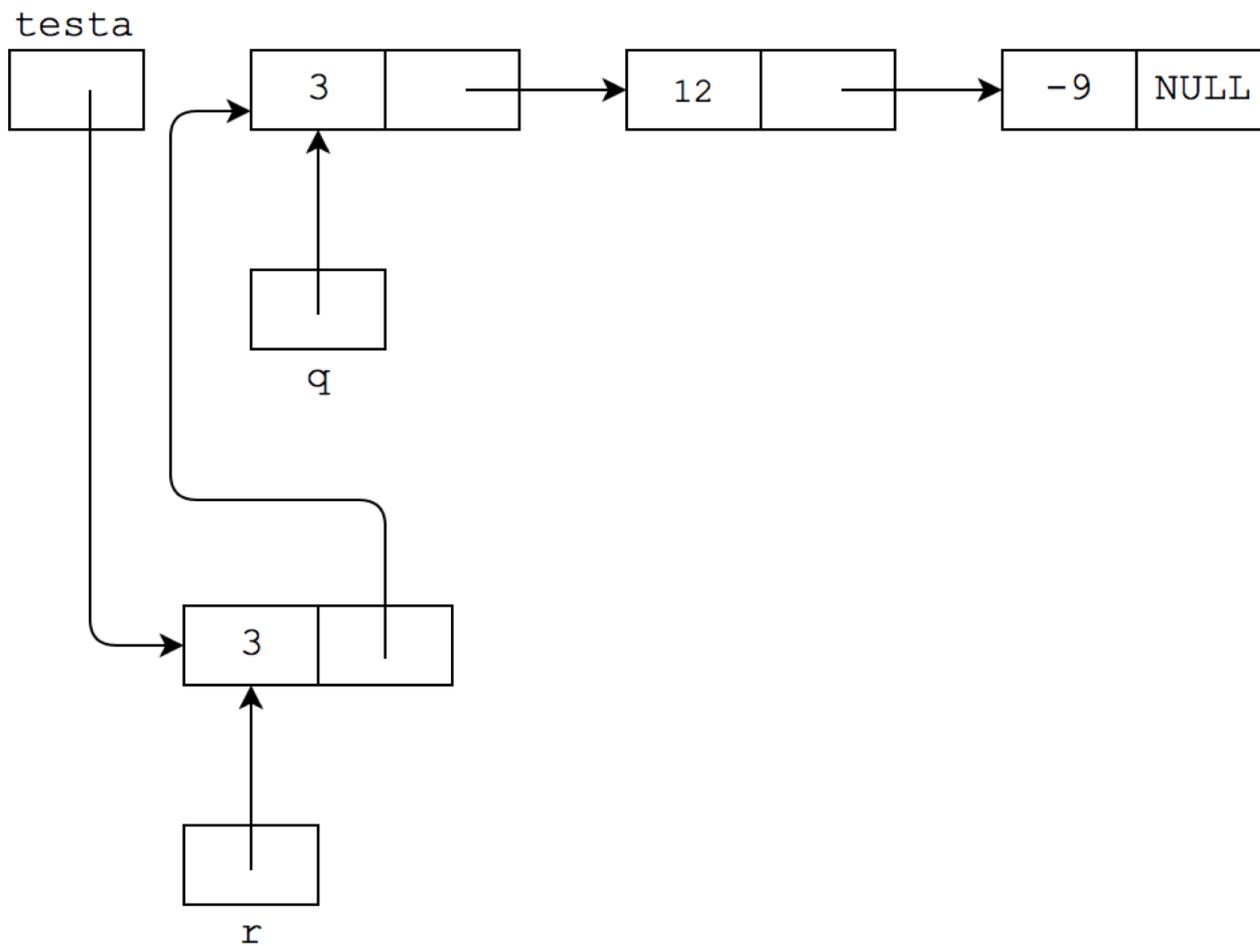
Sì, allora lo dobbiamo duplicare.

Qui il duplicato lo inseriamo a sinistra

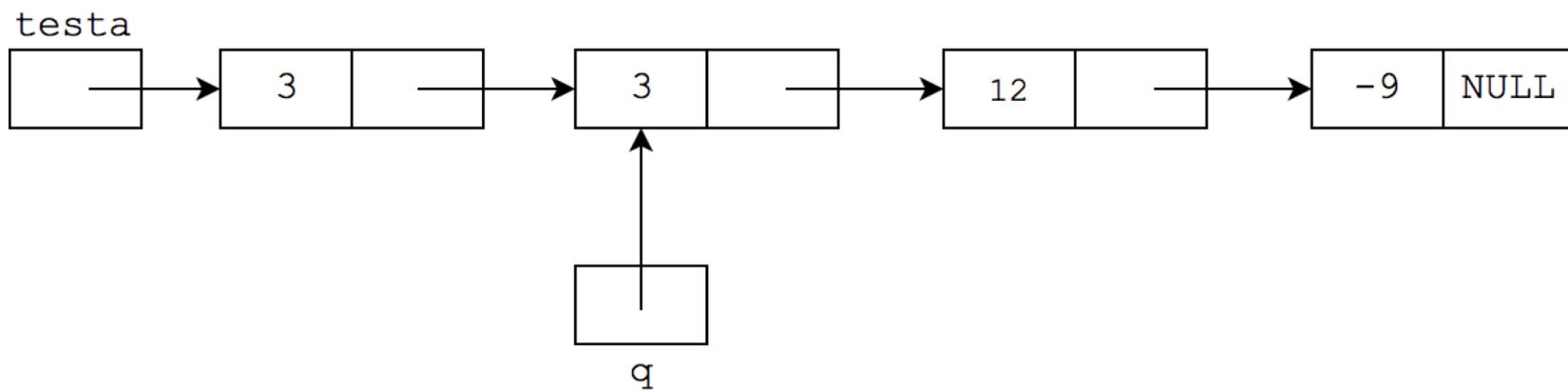
Esercizio 2

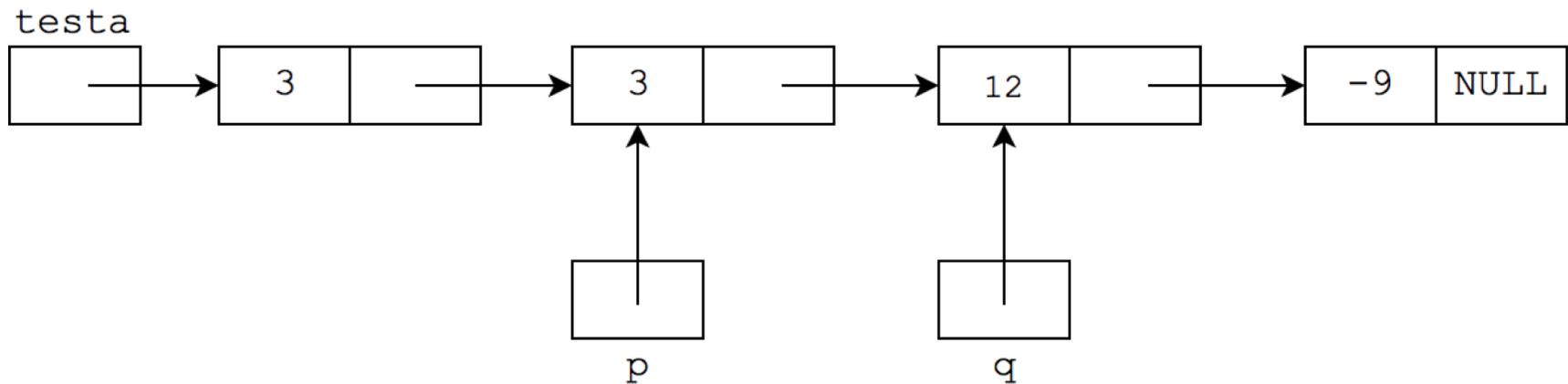


Esercizio 2



Esercizio 2

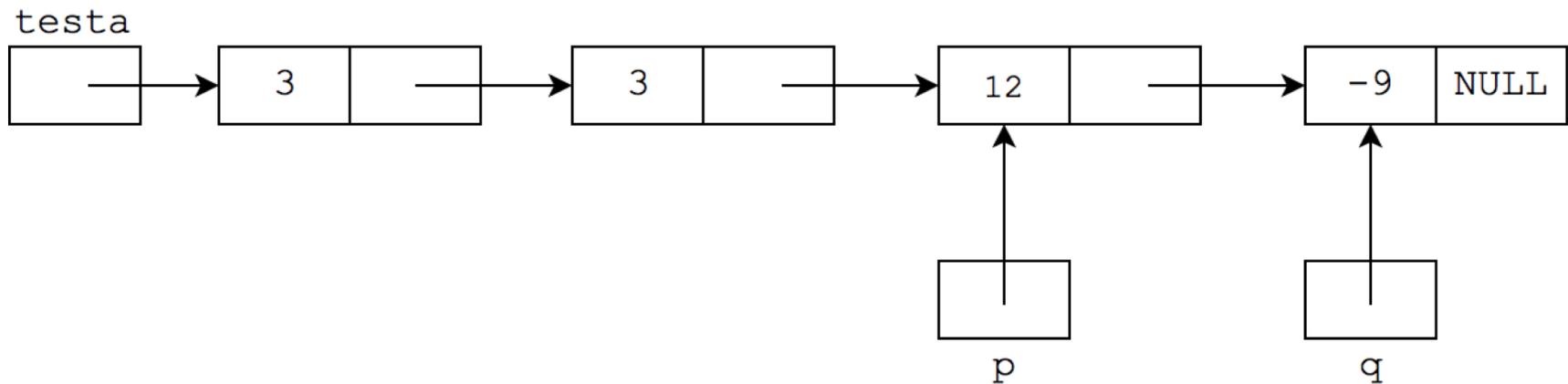




condizione cardine

$q \rightarrow \text{info} < 5?$

No, allora scorriamo sia p che q

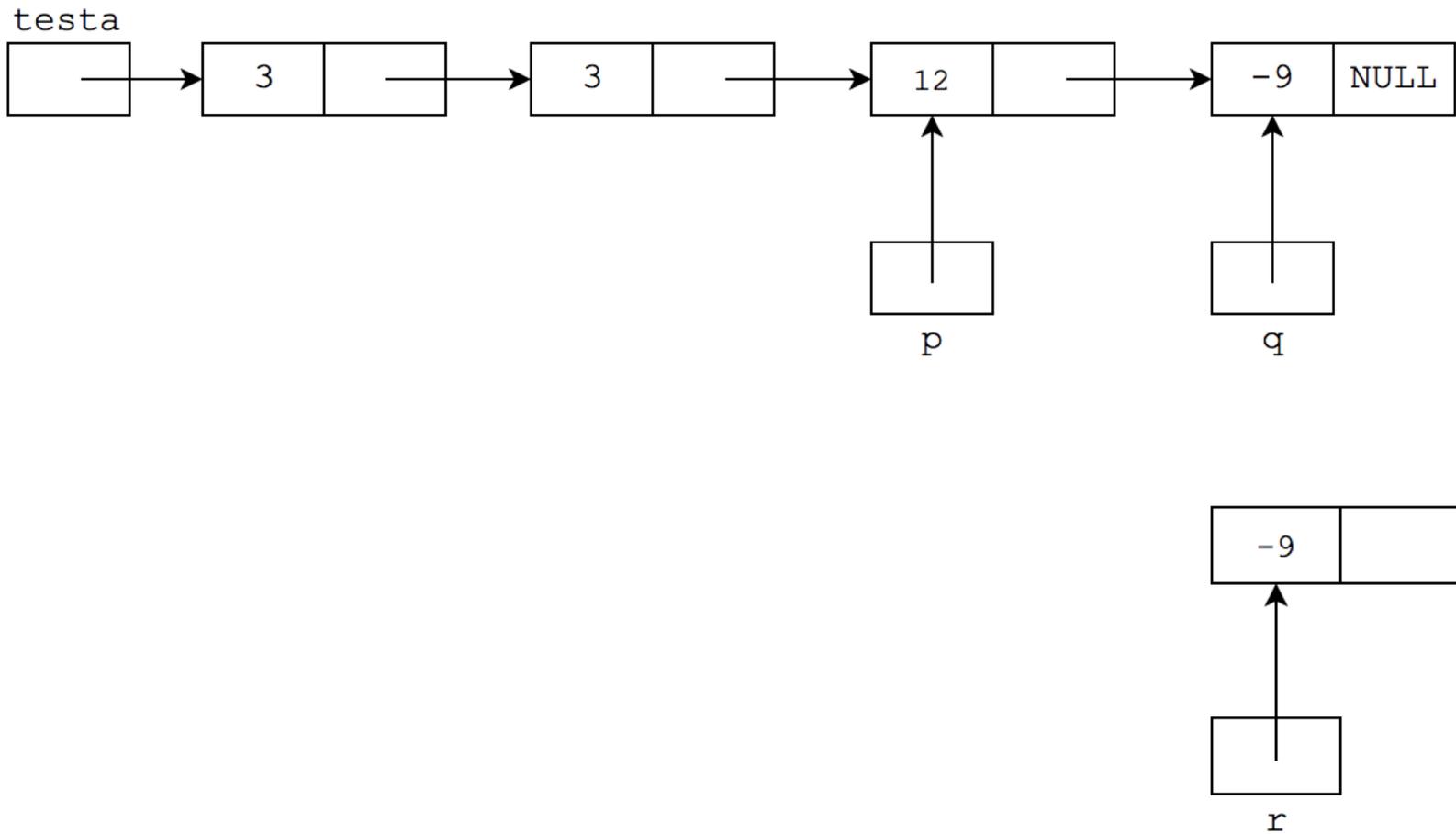


condizione cardine

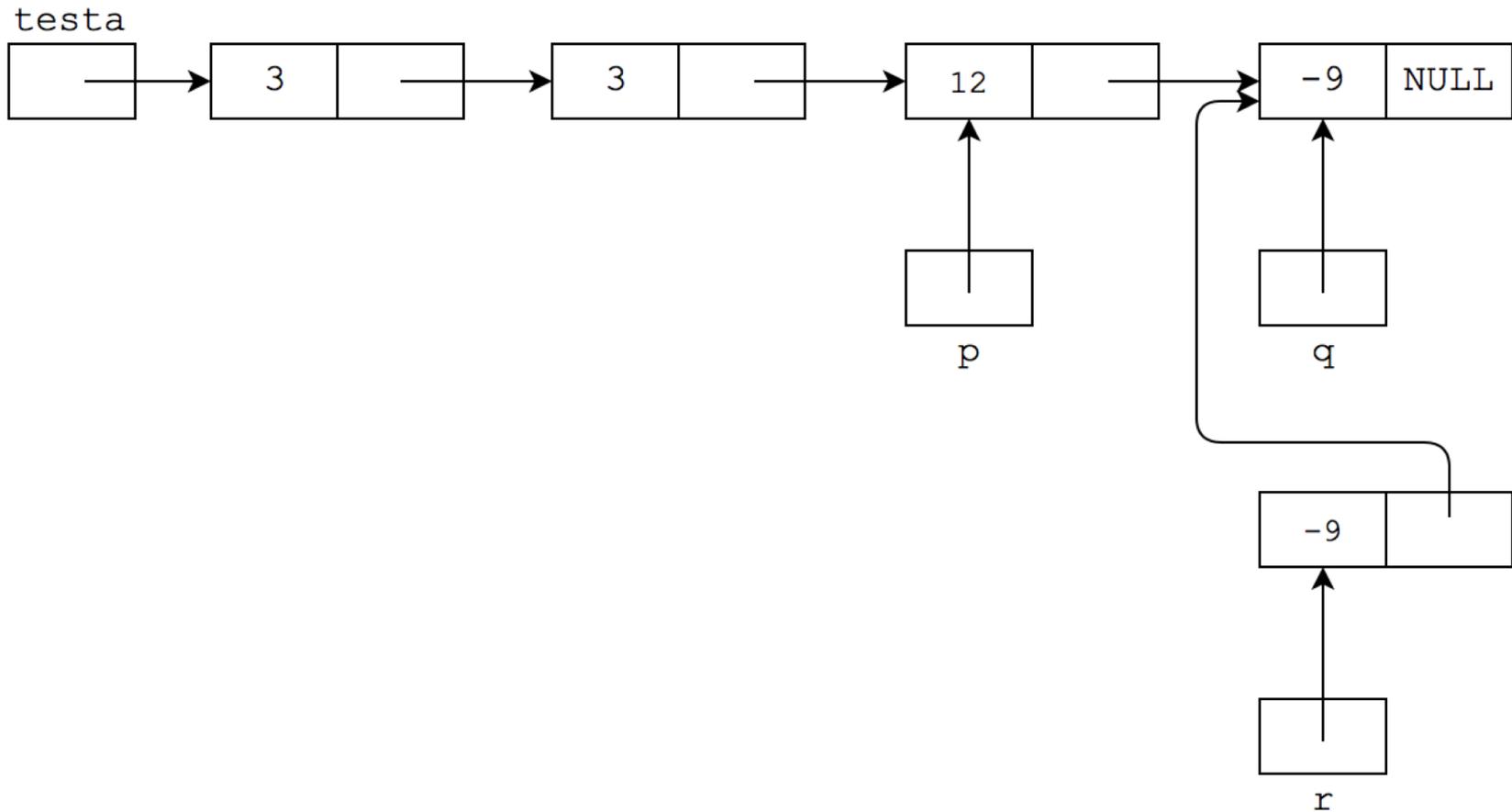
$q \rightarrow \text{info} < 5?$

Sì, allora lo duplichiamo

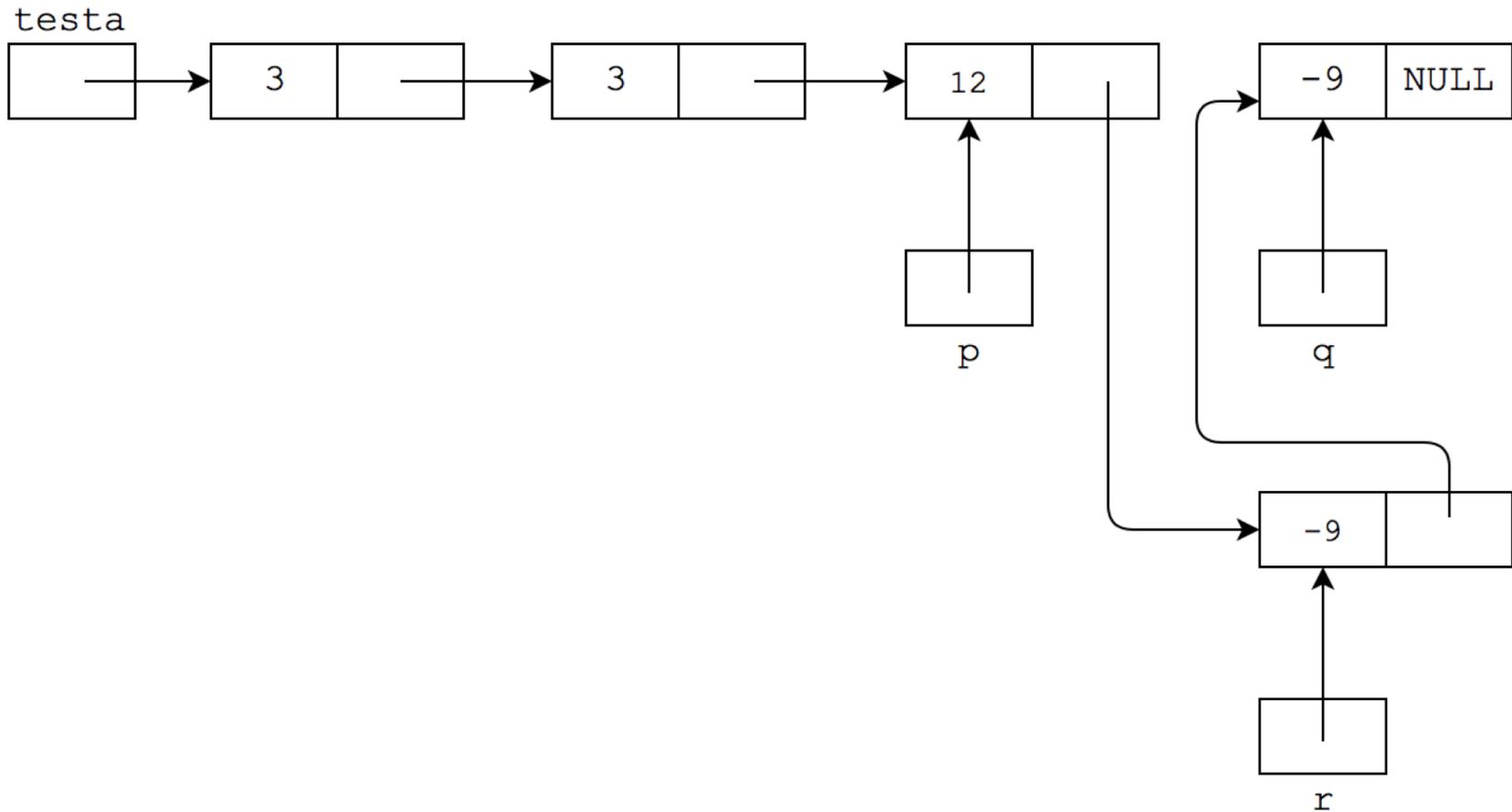
Esercizio 2



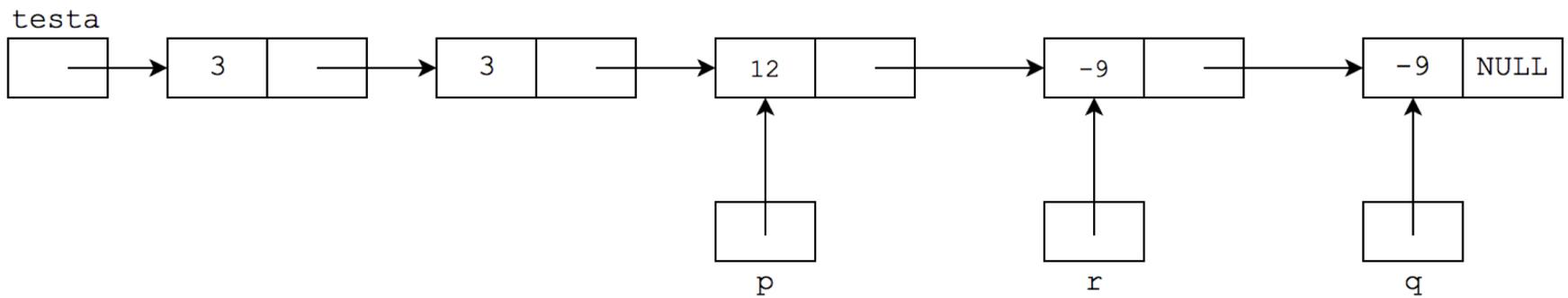
Esercizio 2

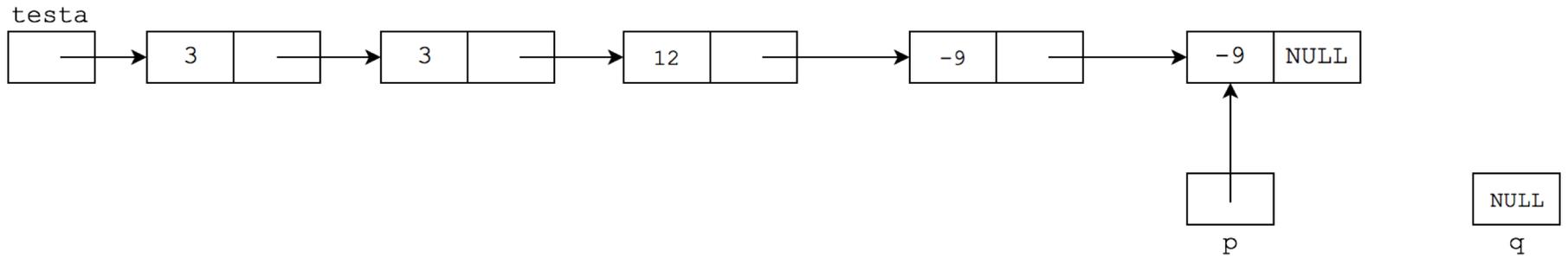


Esercizio 2



Esercizio 2





q diventa NULL, quindi abbiamo scorso tutta la lista,
e non c'è più nulla da fare sulla lista