

Stack

Struttura dati *FILO* (First In Last Out)

Lista

- Push() Aggiunto in testa O(1)

```
void stack_push(my_stack* s, int v){  
    list_insert_front((list_t*)s,v);  
}
```

- Pop() Rimosso dalla testa (delete front) O(1)

```
int stack_pop(my_stack* s){  
    if (s->head!=NULL){  
        int v=s->head->val;  
        list_delete_front((list_t*)s);  
        return v;  
    }
```

Il vantaggio di utilizzare una lista è la locazione dinamica (frammentata)

Array

- Push() Aggiungo in coda O(1)
- Pop() Rimuovo dalla coda O(1)

Il vantaggio degli array sono gli accessi diretti.

Operazione di Top

Top con le liste

```
int stack_top(my_stack* s){  
    if (s->head!=NULL)  
        return s->head->val;  
    printf("ERRORE: stack vuoto!\n");  
    return -1;  
}
```

Notazione

Notazione **Postfissa**

- Espressione ::= N | E E OP

Detta anche notazione polacca inversa. Nella notazione polacca inversa, detta anche notazione postfissa in contrasto con la normale notazione infissa, prima si inseriscono gli operandi e poi gli operatori: un esempio di RPN è 3 2 + che equivale al classico 3+2, oppure 10 2 / che fornisce 5.

esempio 1 2 3 * +

Notazione **Infissa**: (quella che usiamo di solito)

- Espressione ::= N | (E OP E)

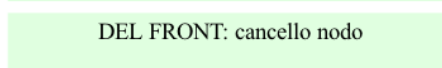
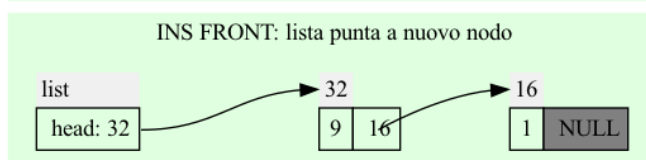
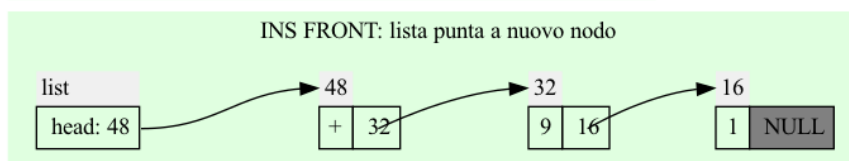
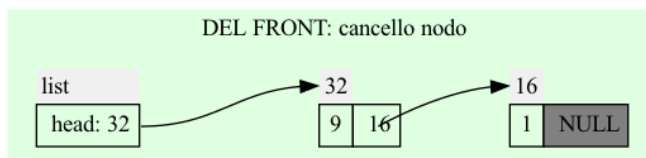
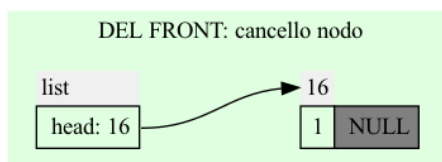
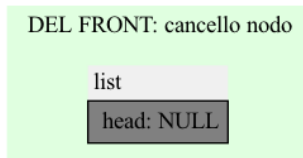
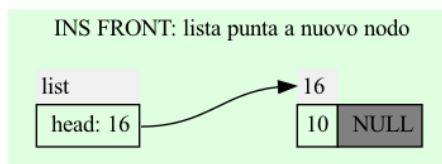
Possiamo ricreare una notazione infissa tramite una stampa di un albero, prima chiamiamo ricorsivamente a sinistra, poi ricorsivamente a destra e infine l'operazione. È quella che si usa di solito per rappresentare le espressioni in un linguaggio di programmazione.

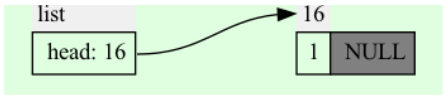
esempio 1 + 2 * 3

La notazione Postfissa permette di utilizzare un approccio **bottom up** e **top down** (file 2-arith-bottom-up-dot.c e 3-arith-top-down-dot.c)

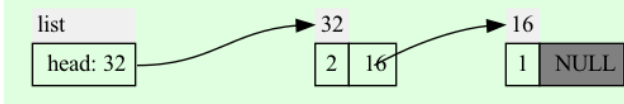
Bottom up:

Esempio con grafico Top-Down: 1 2 3 4 + + +

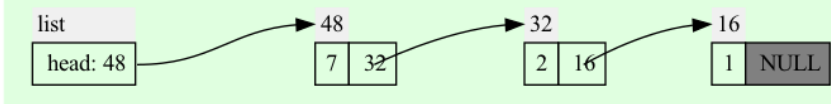




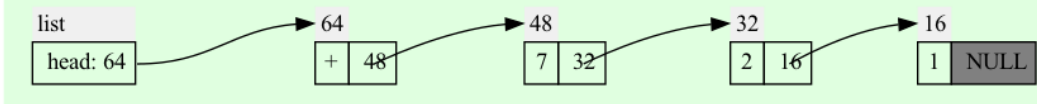
DEL FRONT: cancello nodo



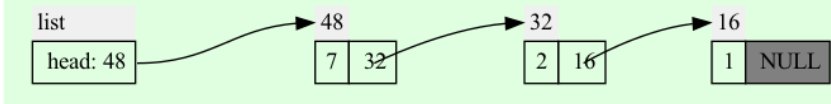
DEL FRONT: cancello nodo



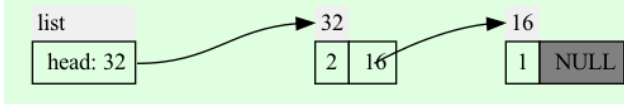
INS FRONT: lista punta a nuovo nodo



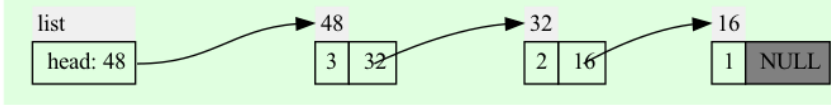
INS FRONT: lista punta a nuovo nodo



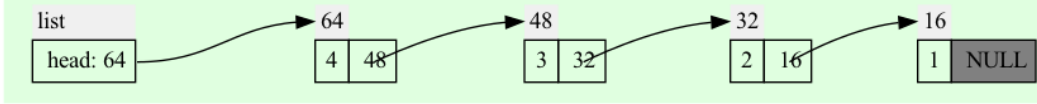
DEL FRONT: cancello nodo



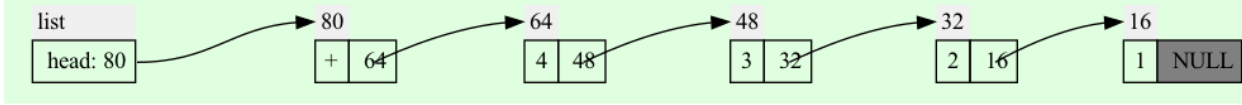
DEL FRONT: cancello nodo



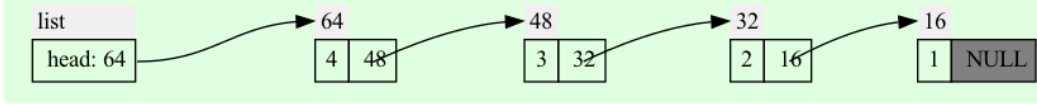
DEL FRONT: cancello nodo



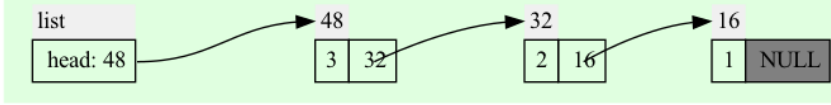
INS FRONT: lista punta a nuovo nodo



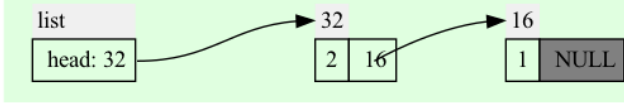
INS FRONT: lista punta a nuovo nodo



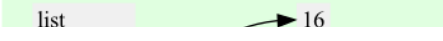
INS FRONT: lista punta a nuovo nodo

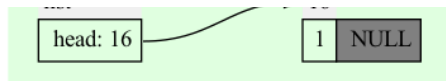


INS FRONT: lista punta a nuovo nodo



INS FRONT: lista punta a nuovo nodo





Top down: Inserisco tutto nello Stack e quando ho finito vado a fare la valutazione di tutto lo stack della chiamata ricorsiva

```
Rec (S)
  Top(S)= op
  M1 = Rec(S)
  M2 = Rec)(S)

  return M1 op M2
```