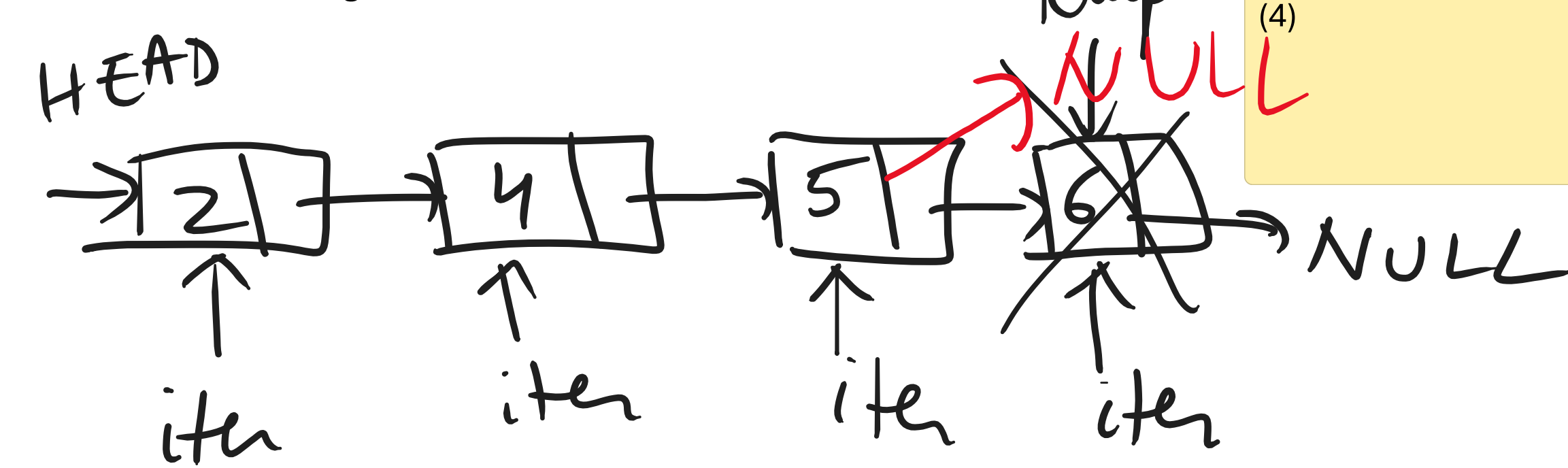
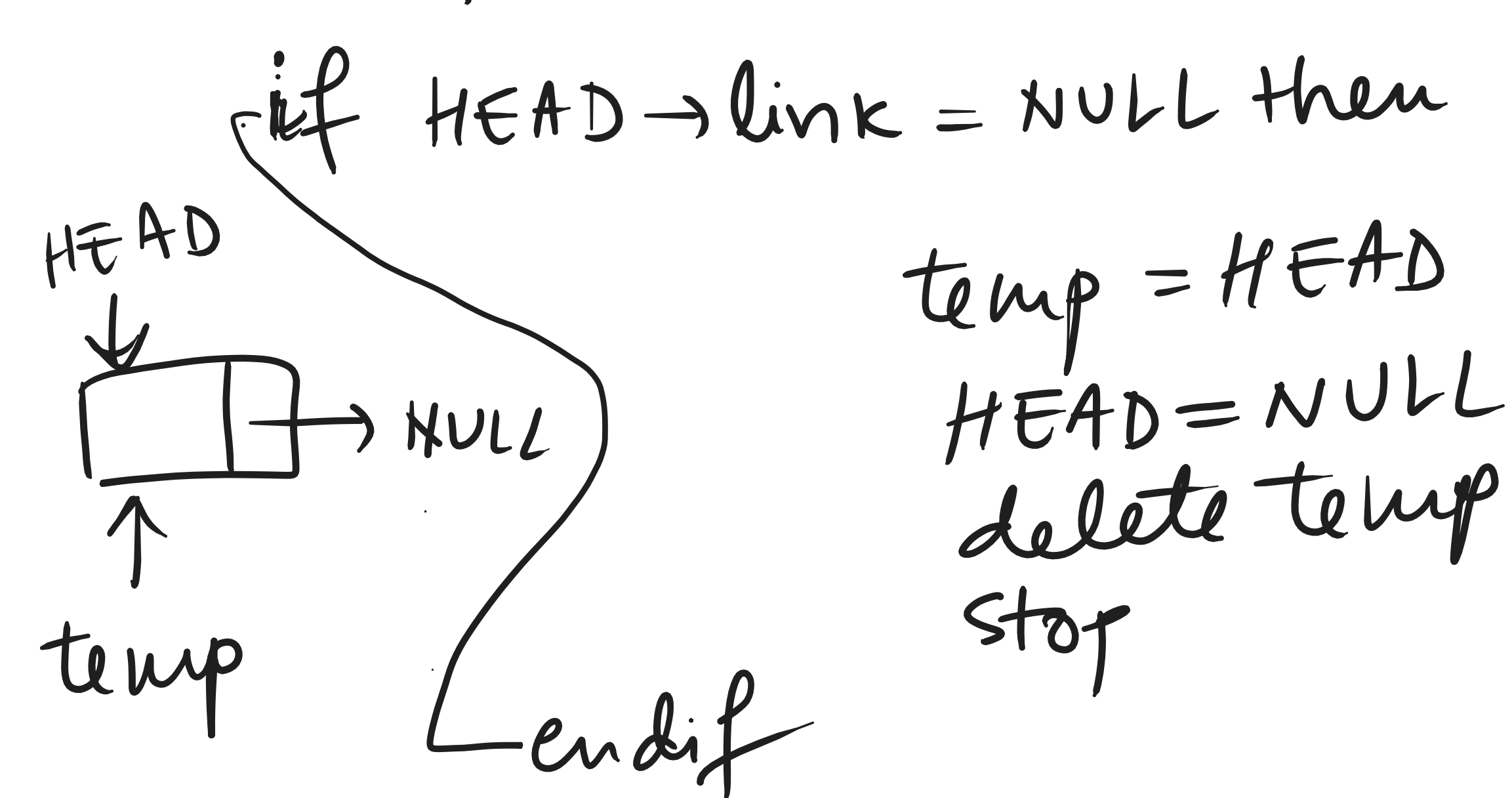


3.2 stergerea ultimului element



```
if HEAD = NULL then write 'UNDERFLOW'
stop
endif
```



```
temp = HEAD
HEAD = NULL
delete temp
stop

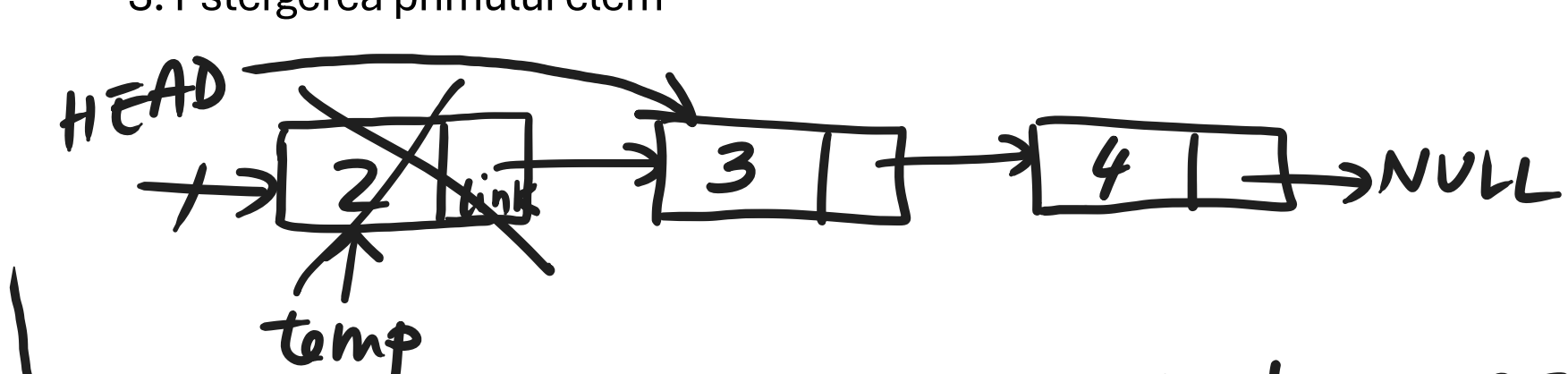
iter = HEAD
while (iter -> link) -> link != NULL
    iter = iter -> link
endwhile
```

```
temp = iter -> link
iter -> link = NULL
delete temp
```

Liste intantuite

Stergerea unui element

3.1 stergerea primului elem

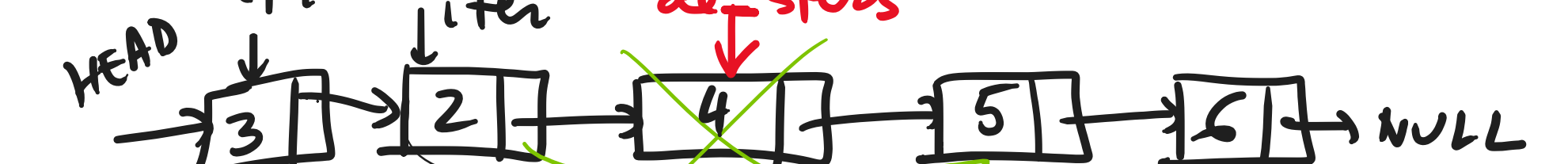


```
if HEAD = NULL then write 'UNDERFLOW'
stop
endif
temp = HEAD
HEAD = HEAD -> link
Eliberază zona de memorie punctată de temp.
(In C++ : delete temp;)
```

Funcționează dacă este un singur elem? DA

3.3 Stergerea unui elem dat:

Se da de_sters pointer la elementul care trebuie sters.
Presupunem ca de_sters != NULL si puncteaza la un elem dat



```
if HEAD = NULL then write 'UNDERFLOW'
stop
endif
if de_sters = NULL then write 'pozitie invalida'
stop
endif
iter = HEAD
while (iter -> link != de_sters)
    iter = iter -> link
endwhile
iter -> link = de_sters -> link
delete de_sters
```

Pt a functiona dacă de_sters este la primul elem

```
if de_sters = HEAD then apelează stergere-inceput
stop
endif
```

Algoritmi:

inserarea unui nod nou	stergerea/accesarea unui nod
<pre>Aloca memorie pentru un nod nou. Returneaza p, un pointer la noul nod. if p = NULL then write OVERFLOW stop endif p -> link = top p -> info = a top = p</pre>	<pre>if top = NULL then write UNDERFLOW stop endif elem_sters = top -> info temp = top top = top -> link delete temp</pre>

