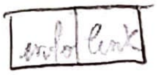
Liste Inlantuite

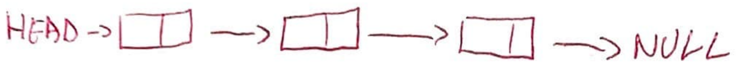
Fiecare element:



```
STRUCT NOD{
    T info;
  NOD *link;  -> adresa de memorie a urmatorului element
};
```

Orice listă este bine definită dacă se cunoaște

HEAD = pointer la primul element

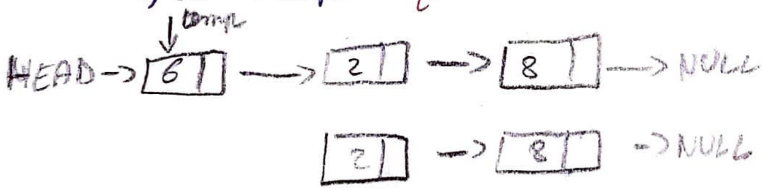


Operatii de bază:

- accessarea / modificarea
- inserarea
  1) inceput
  2) sfarsit
  3) după un element

} Dată Precată

- stergerea unui element
  1) La inceput {NU MERGE CÂND ESTE UNDERFLOW} HEAD = NULL

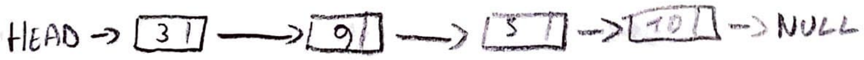


temp = HEAD

HEAD = HEAD->link

Eliberează zona dem memorie care punctează temp. delete temp;

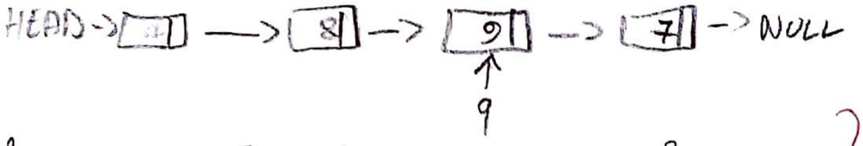
2) La sfarsit {NU MERGE CÂND ESTE UNDERFLOW}



```
iter = HEAD
while iter ≠ NULL and iter->link->link and iter->link
|     iter = iter->link
endwhile
delete iter->link;  ]
iter->link = NULL;  ] if iter -> link ≠ Null
delete iter;        ]
HEAD = NULL;        ] else
```

3) După un element dat

Ex. se dă q pointer la elementul care trebuie șters; Se presupune că q ≠ NULL și punctează la un element din listă



①

```
if HEAD = NULL { cout "underflow"; stop;}
iter = HEAD
At while iter ≠ NULL && iter -> link ≠ q {
        iter = iter -> link;
}
if iter = NULL { cout "Nu există q"; stop;}
iter -> link = q -> link;
delete q;
```

NU MERGE PENTRU STERGEREA LA INCEPUT și la SFARȘIT