

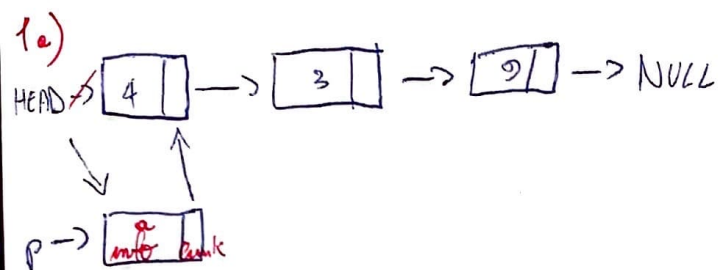
Operații de listă

1) Inserarea unui element

a) la început

b) la sfârșit

c) după un element dat



Alocă zona de memorie pt un nou nod. Fie p pointer la zona de memorie.

```

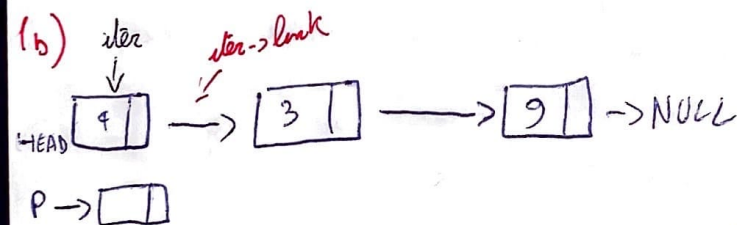
if p = NULL // nu există zona de memorie liberă
then write "OVERFLOW"
stop
endif
  
```

Funcționează și pentru HEAD = NULL (listă vidă)

$p \rightarrow \text{info} = a;$

$p \rightarrow \text{link} = \text{HEAD};$

$\text{HEAD} = p;$



Alocă zona de memorie pt nou nod. // C++ $\rightarrow p = \text{new Node}$

```

if p = NULL // nu există zona
then "OVERFLOW"
stop
endif
  
```

$p \rightarrow \text{info} = a;$

$p \rightarrow \text{link} = \text{NULL};$

$\text{iter} = \text{HEAD};$

While $\text{iter} \rightarrow \text{link} \neq \text{NULL}$ && $\text{iter} \neq \text{NULL}$

$\text{iter} = \text{iter} \rightarrow \text{link};$

end While

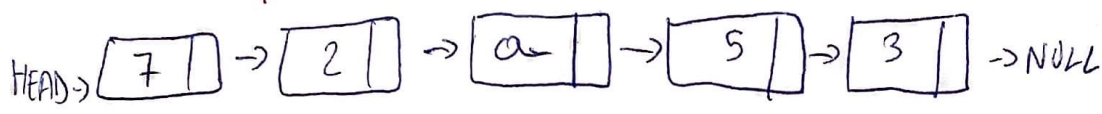
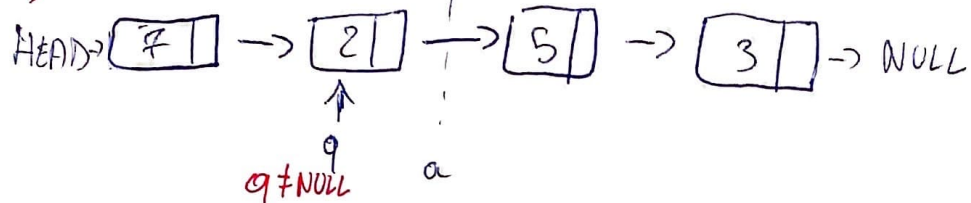
if $\text{iter} \neq \text{NULL}$ else

then $\text{iter} \rightarrow \text{link} = p$ HEAD = p

endif

Funcționează și pentru HEAD = NULL

10)



Alocă zonă de memorie un nou nod (p = new Node,)

```
if p=NULL
    then write "OVERFLOW"
    stop
endif

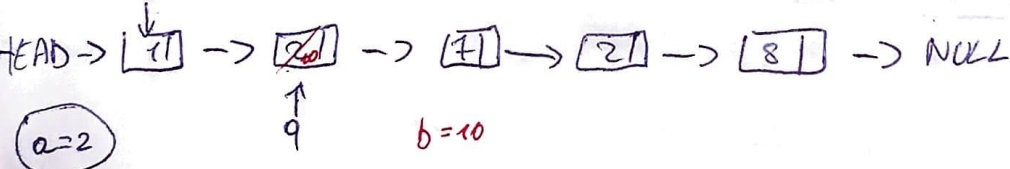
p->info = a
p->link = q->link
q->link = p
```

Funcționare și dacă q=HEAD și q=NULL

2) Accessarea unui element

3) Modificarea unui element

Se dă a o valoare care se caută în listă. [Se returnează un pointer la prima valoare găsită sau NULL dacă nu a fost găsită] Nu



```
iter = HEAD;
while iter ≠ NULL and iter->info ≠ a
    iter = iter->link;
endif

if iter ≠ NULL then
    write "element gasit";
    iter->info = b;
else
    write "element negasit";
endif

return iter;
```