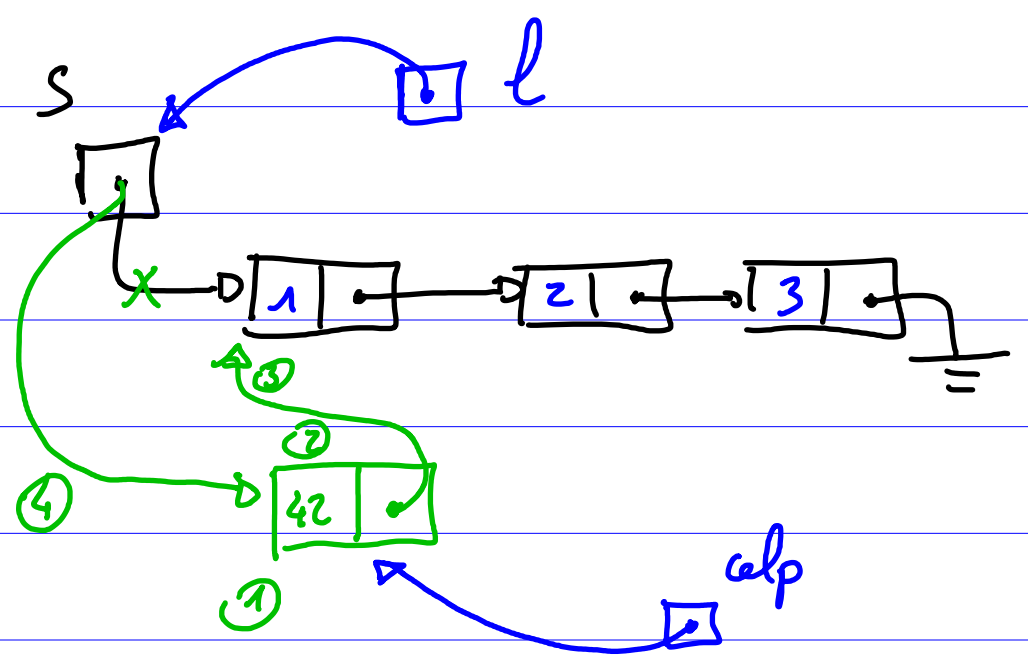
ajout_tete (S, x)

cel ← nouvelle cellule

cel.valeur ← x

cel.suivant ← S.tete

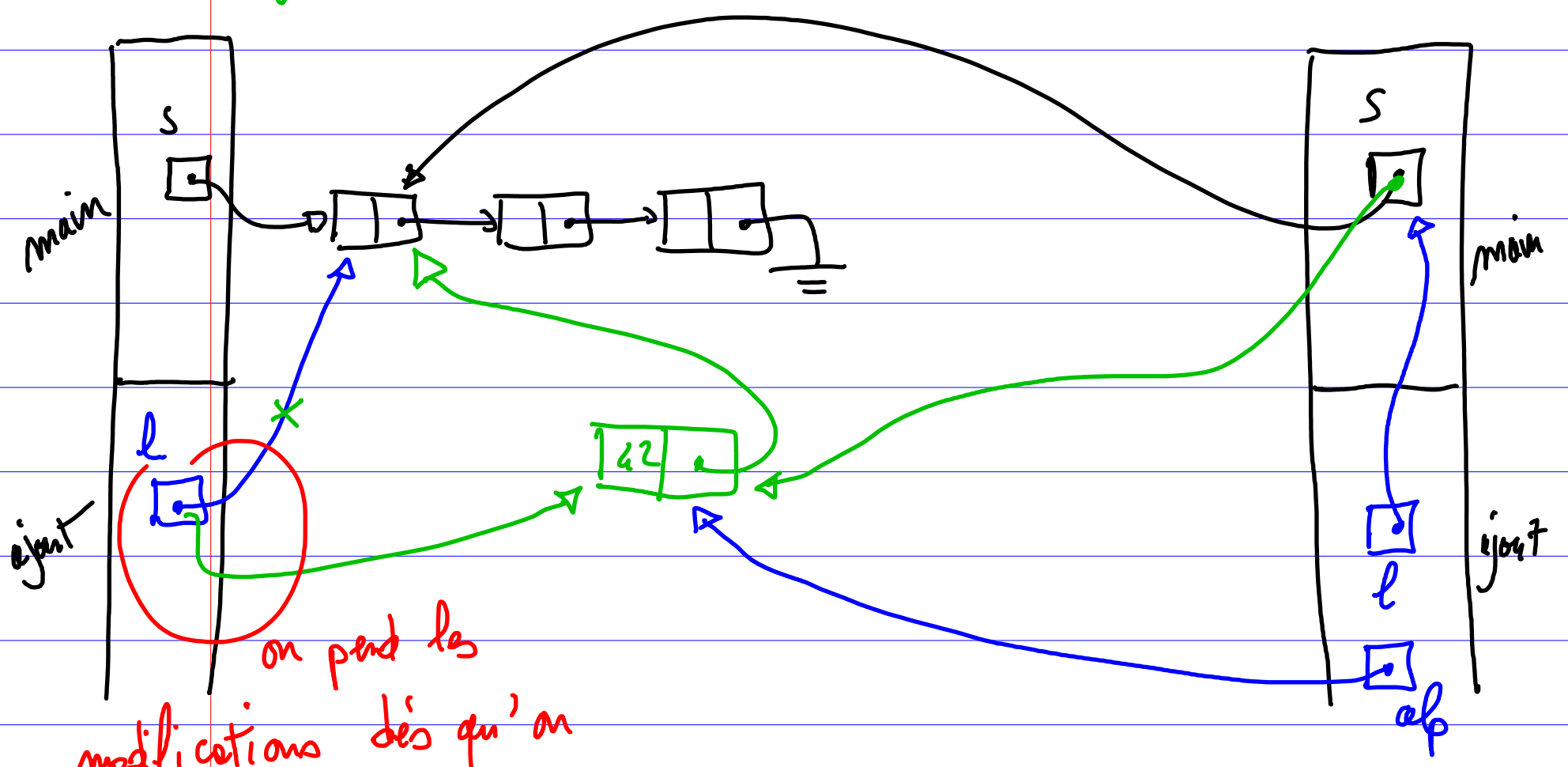
S.tete ← cel




```
void ajout_en_tete (liste_t *l, int n) {
    cellule_t *celp;
    celp = (cellule_t*)malloc (sizeof (cellule_t));
    (*celp).valeur = n;          /* celp->valeur = n; */
    (*celp).suivant = (*l).tete; /* celp->suivant = l->tete */
    (*l).tete = celp;            /* l->tete = celp; */
}
```

Version passe liste_t l

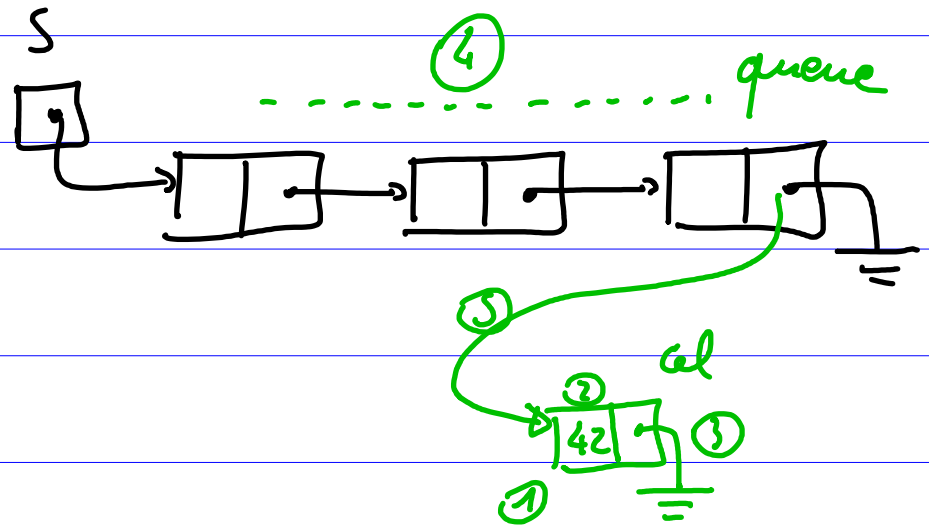
Version correcte liste_t *l



on perd les modifications dès qu'on sort de la fonction !

(et on perd l'adresse de la cellule créée)

ajout_queue (S, x)



Si S.tete = Nil

ajouter_tete (S, x)

retourner

cel ← nouvelle cellule

cel.valeur ← x

cel.suivant ← Nil

queue ← S.tete

Tant que queue.suivant ≠ Nil

queue ← queue.suivant

queue.suivant ← cel

```
void ajout_en_queue (liste_t *l, int n) {
    if (!l->tete) {
        ajout_entete (l, n);
        return;
    }
    cellule_t * cel = (cellule_t*) malloc (sizeof (cellule_t));
    cel->valeur = n;
    cel->suivant = NULL;
    cellule_t * queue = l->tete;
    while (queue->suivant) {
        queue = queue->suivant;
    }
    queue->suivant = cel;
}
```

somme (S)

sum ← 0

cel ← S. tete

Tant que cel ≠ Nil

sum ← sum + cel. valeur

cel ← cel. suivant

retourner sum

cellule_t * cel = l → tete

(comme ajout en queue)