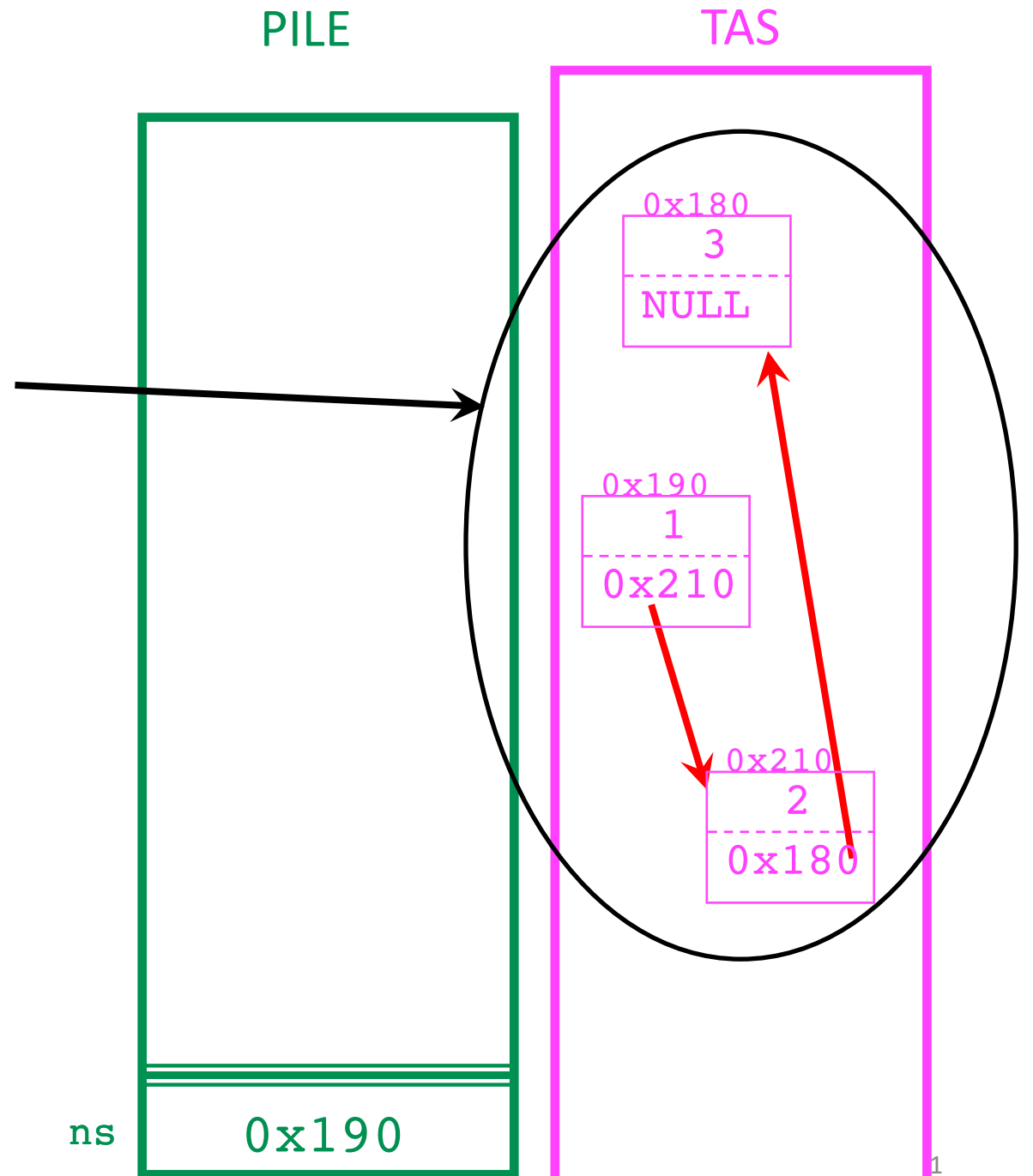


Libérer une liste

L'objectif est de passer de ceci

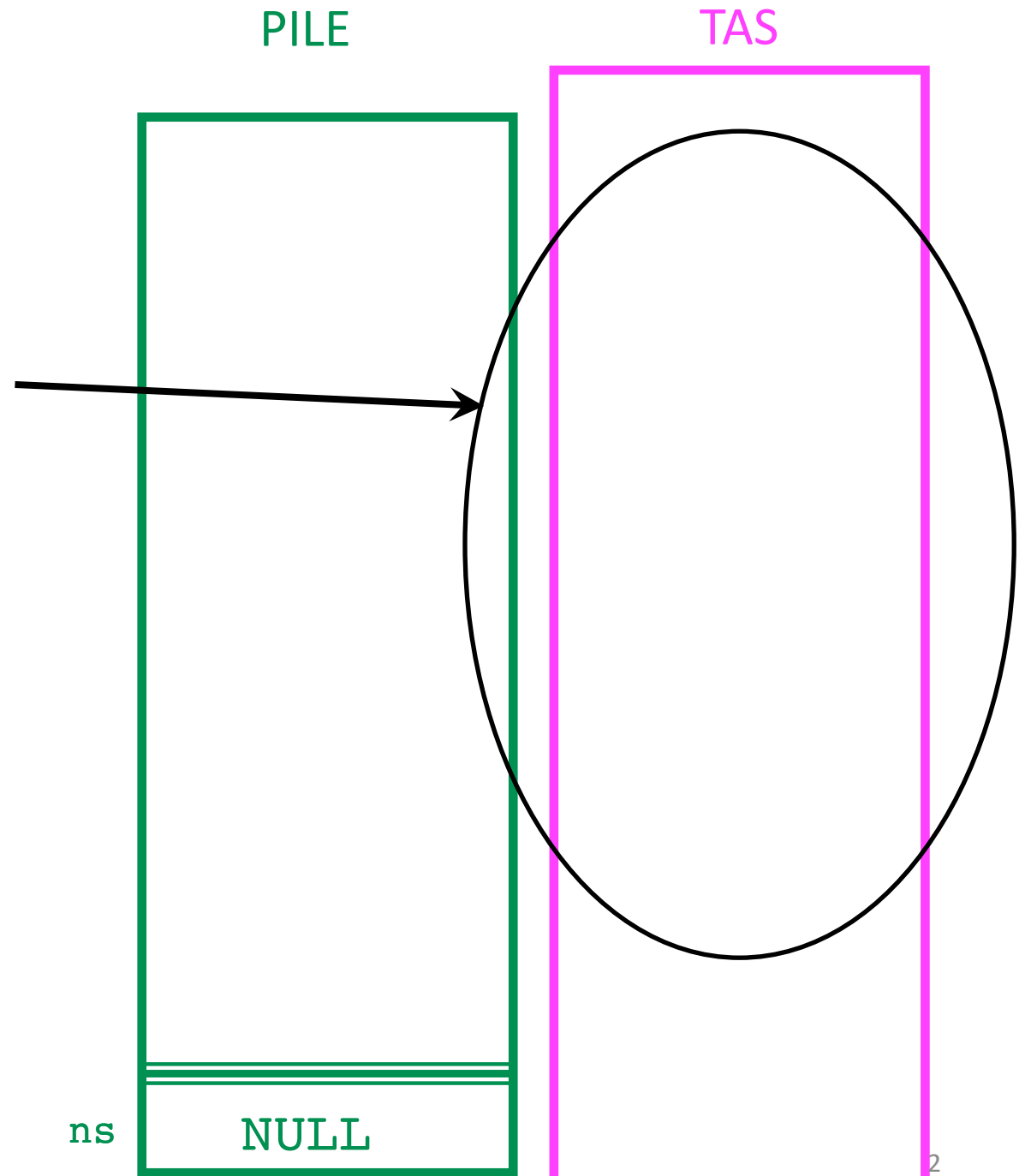
```
int main() {  
    cellule_t * ns;  
    ns=cons(3,ns);  
    ns=cons(2,ns);  
    ns=cons(1,ns);  
    ns=Liberer_liste(ns);  
    return 0;  
}
```



Libérer une liste

```
int main(){  
    cellule_t * ns;  
    ns=cons(3,ns);  
    ns=cons(2,ns);  
    ns=cons(1,ns);  
    ns=Liberer_liste(ns);  
    return 0;  
}
```

à ceci



Libérer une liste

Comment faire ?

Pour chaque élément de la liste

Le libérer (`free`)

Passer au suivant

- Quel est la signature de la fonction ?

```
cellule_t *Liberer_liste(cellule_t *liste);
```

Libérer une liste

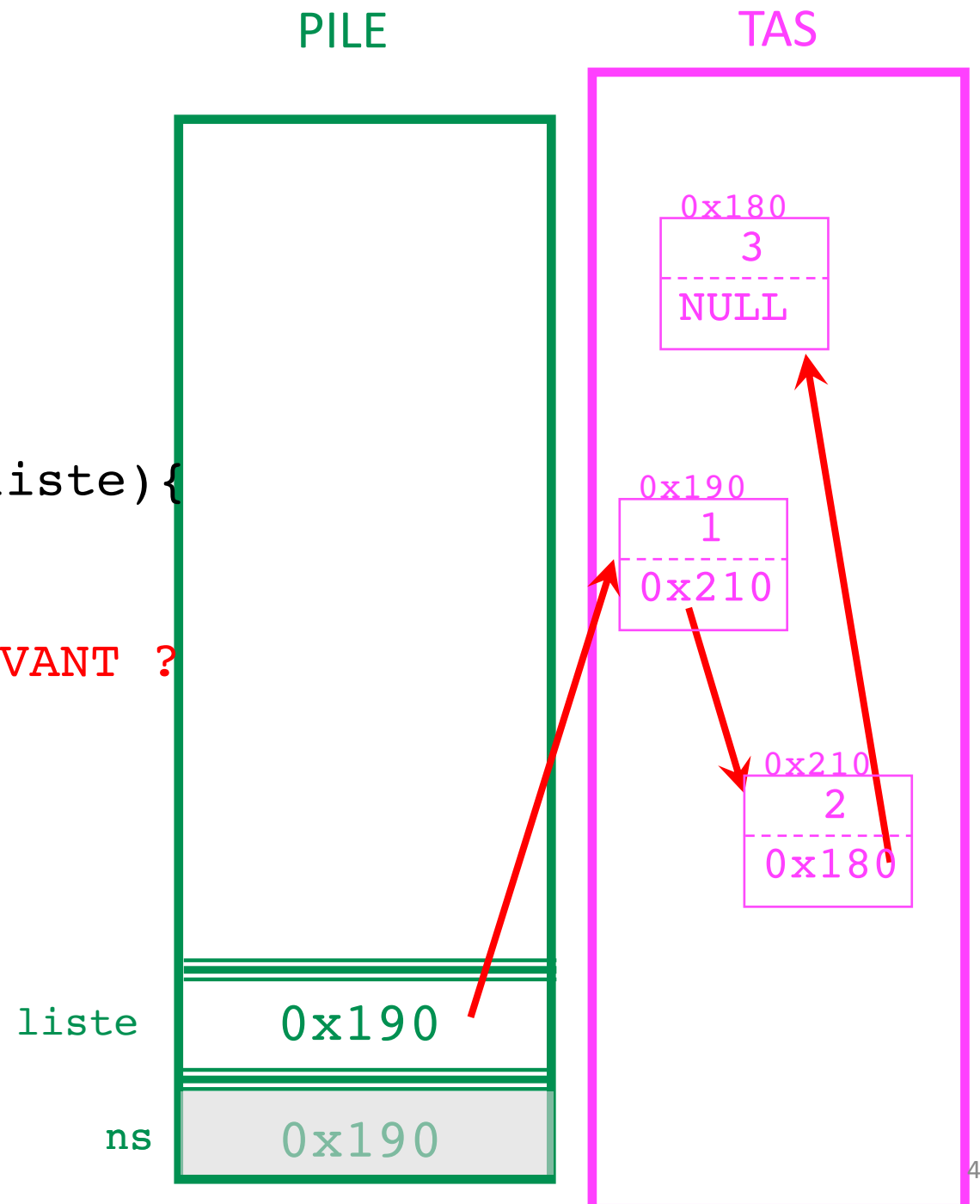
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    → while(liste != NULL){  
        free(liste);  
        MAIS COMMENT PASSER AU SUIVANT ?  
    }  
    return NULL;  
}
```



Libérer une liste

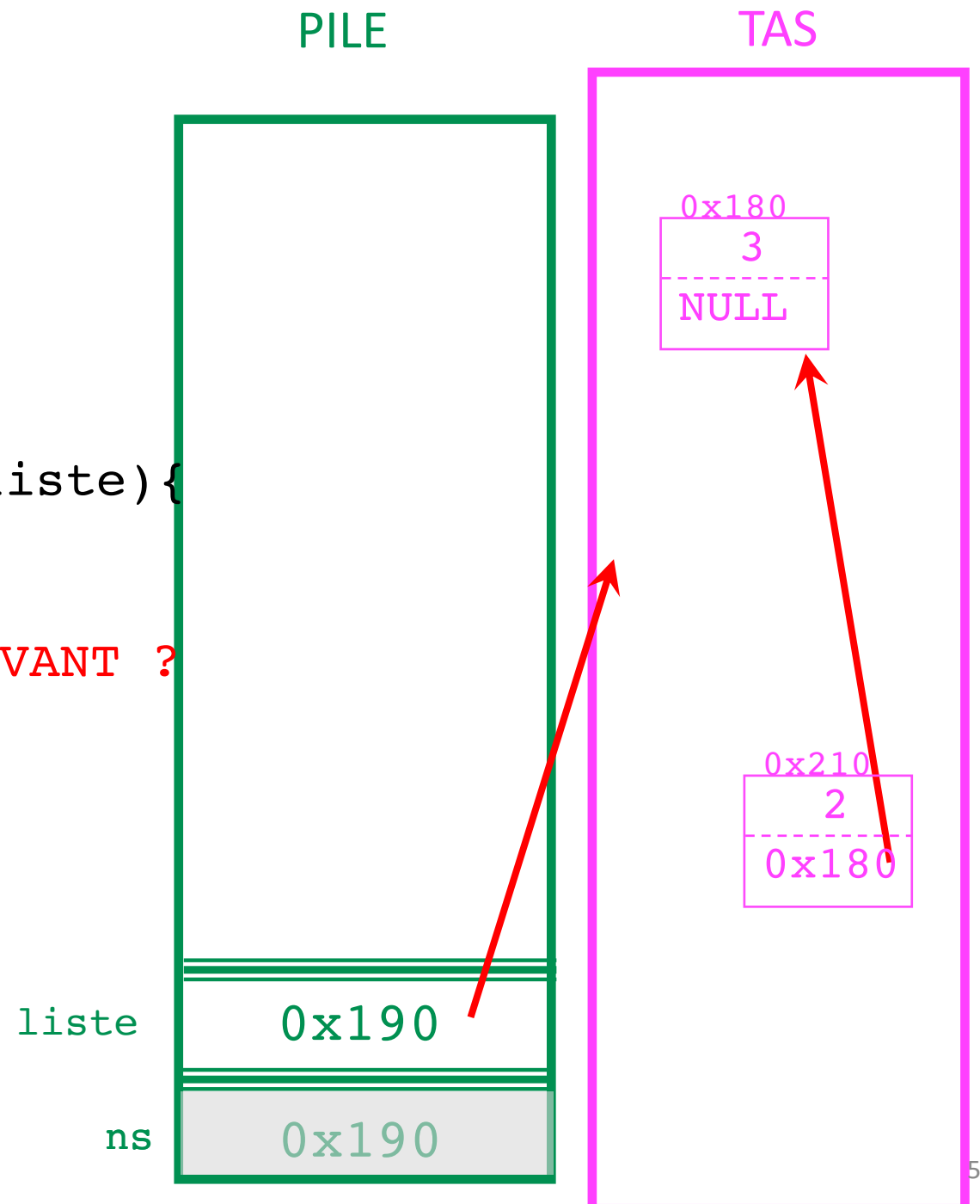
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    while(liste != NULL){  
→       free(liste);  
        MAIS COMMENT PASSER AU SUIVANT ?  
    }  
    return NULL;  
}
```



Libérer une liste

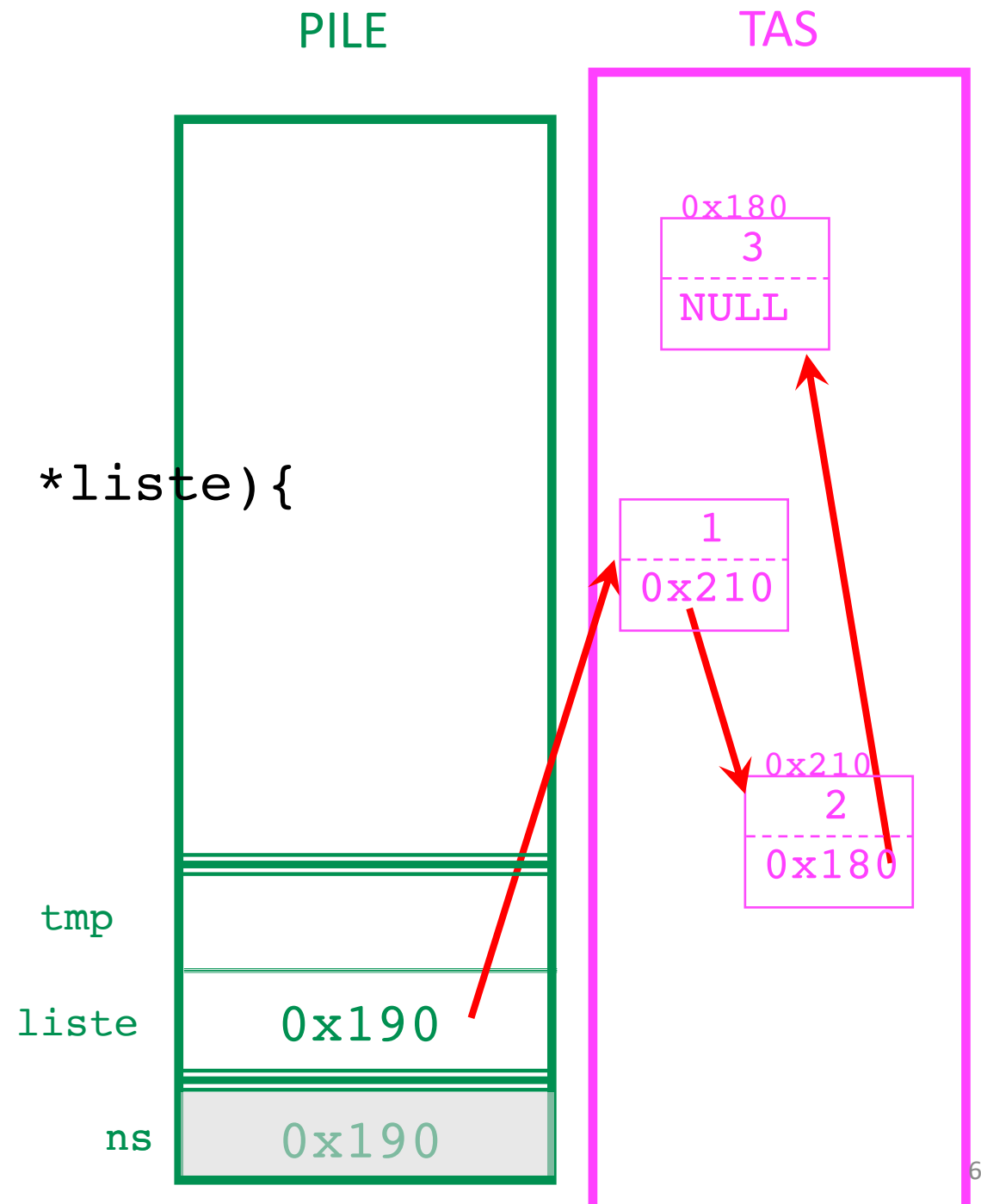
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    → while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

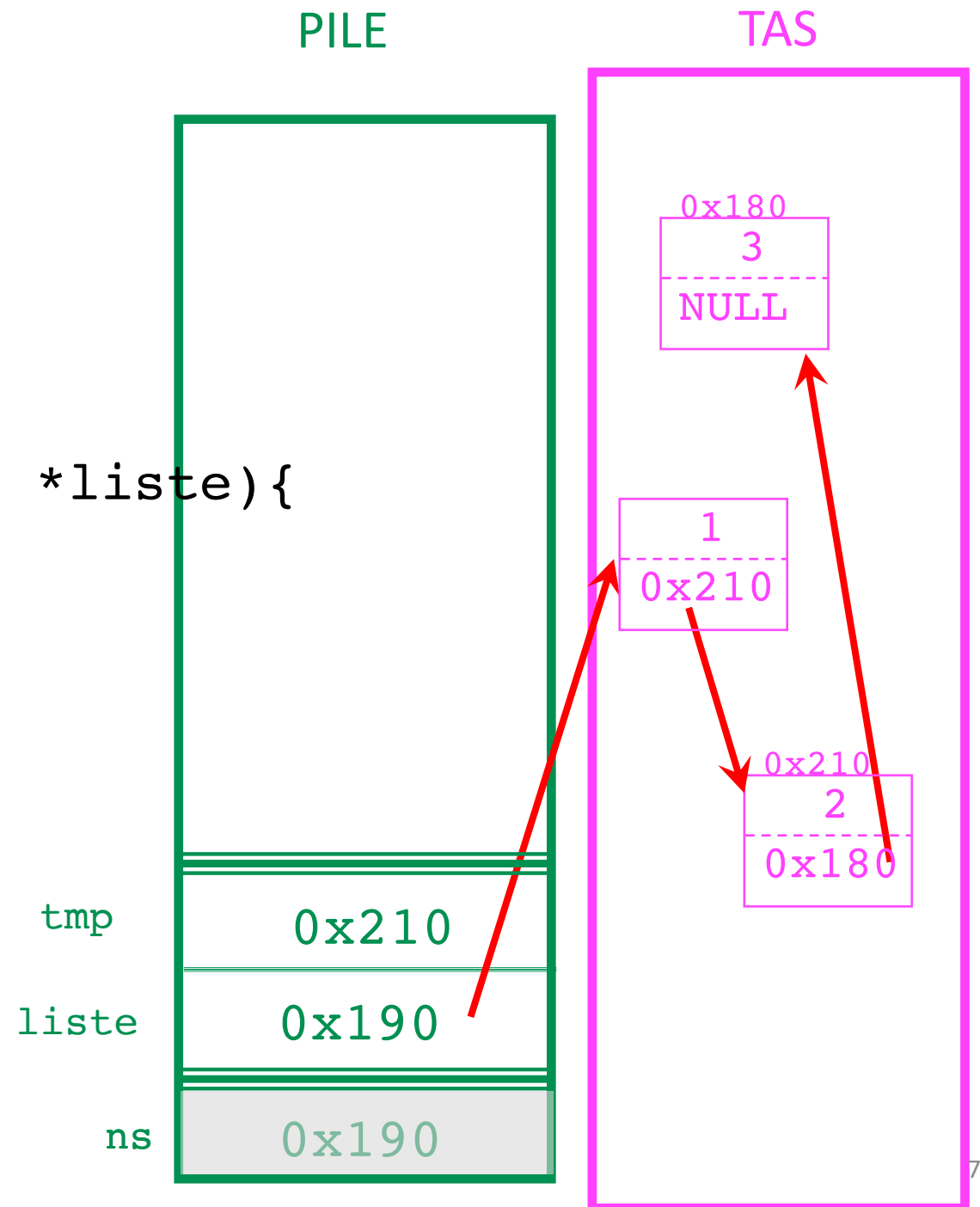
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
→       tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

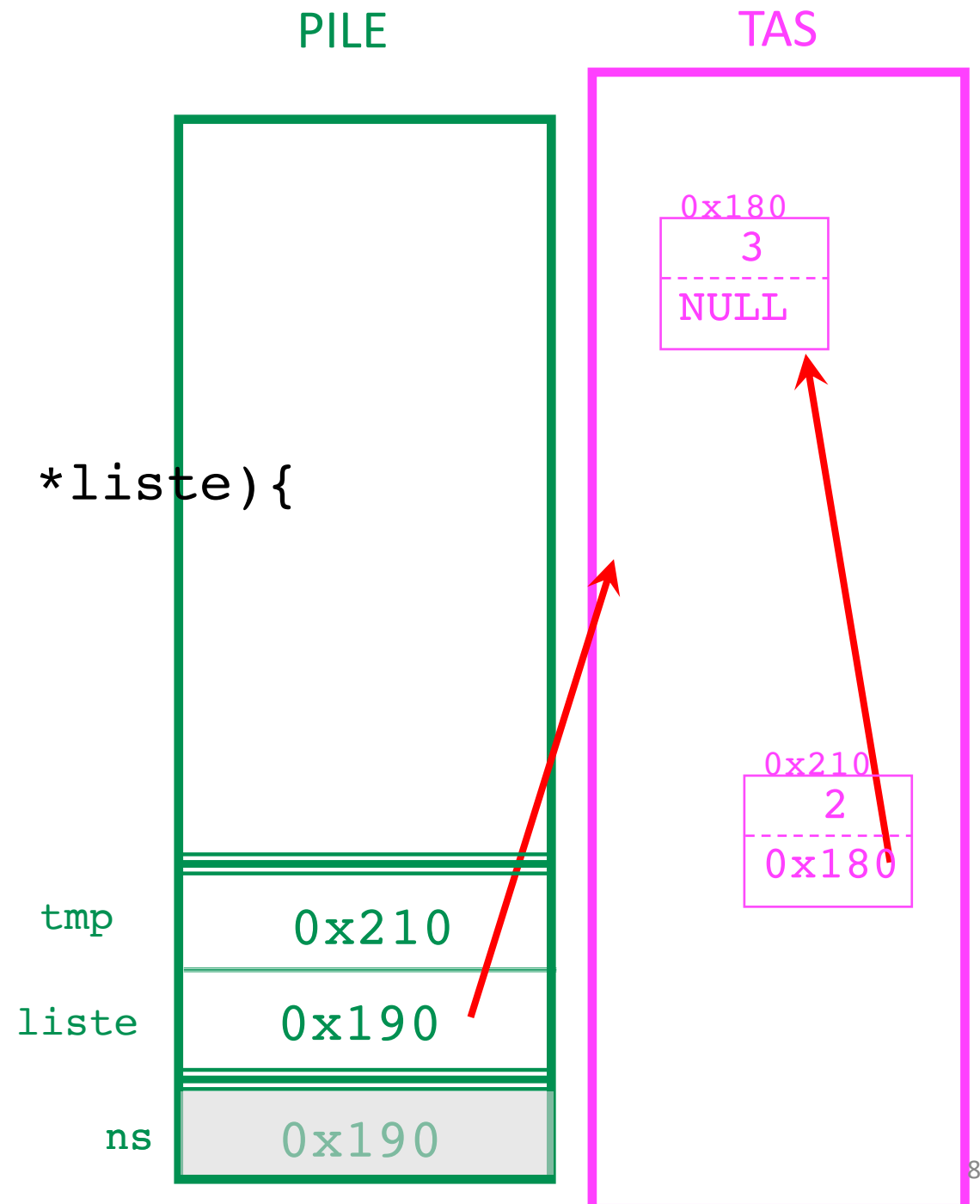
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suitant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

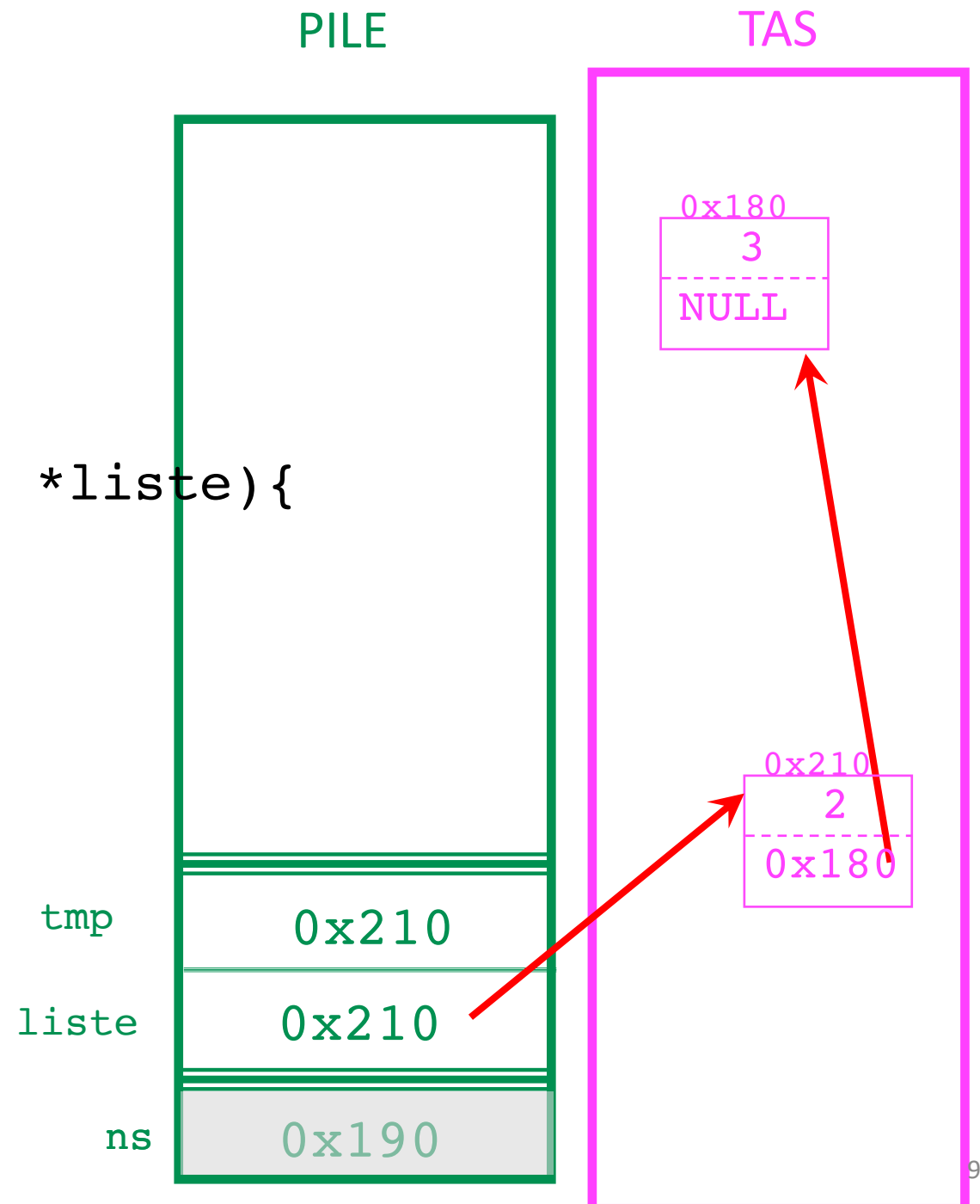
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

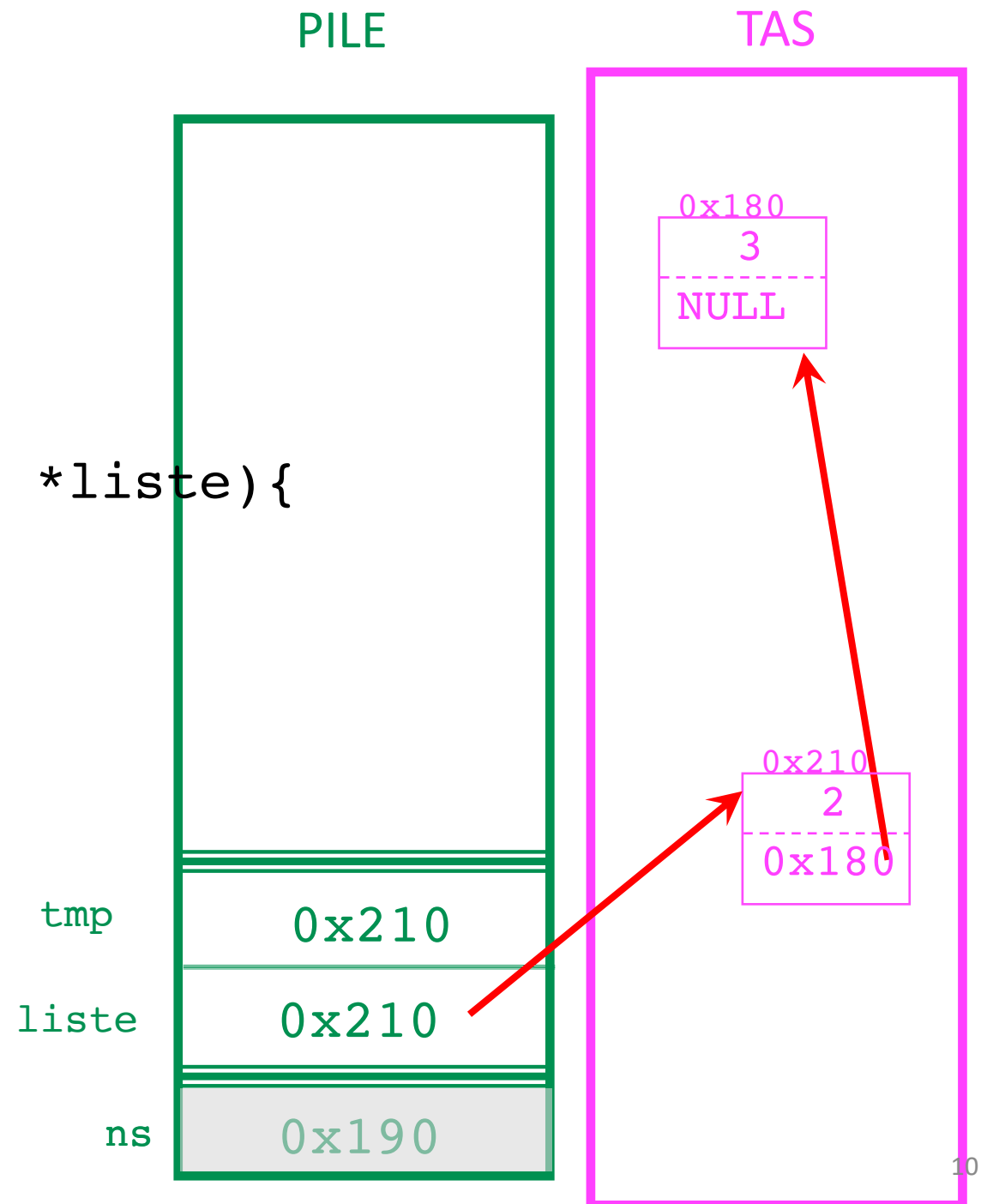
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    → while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

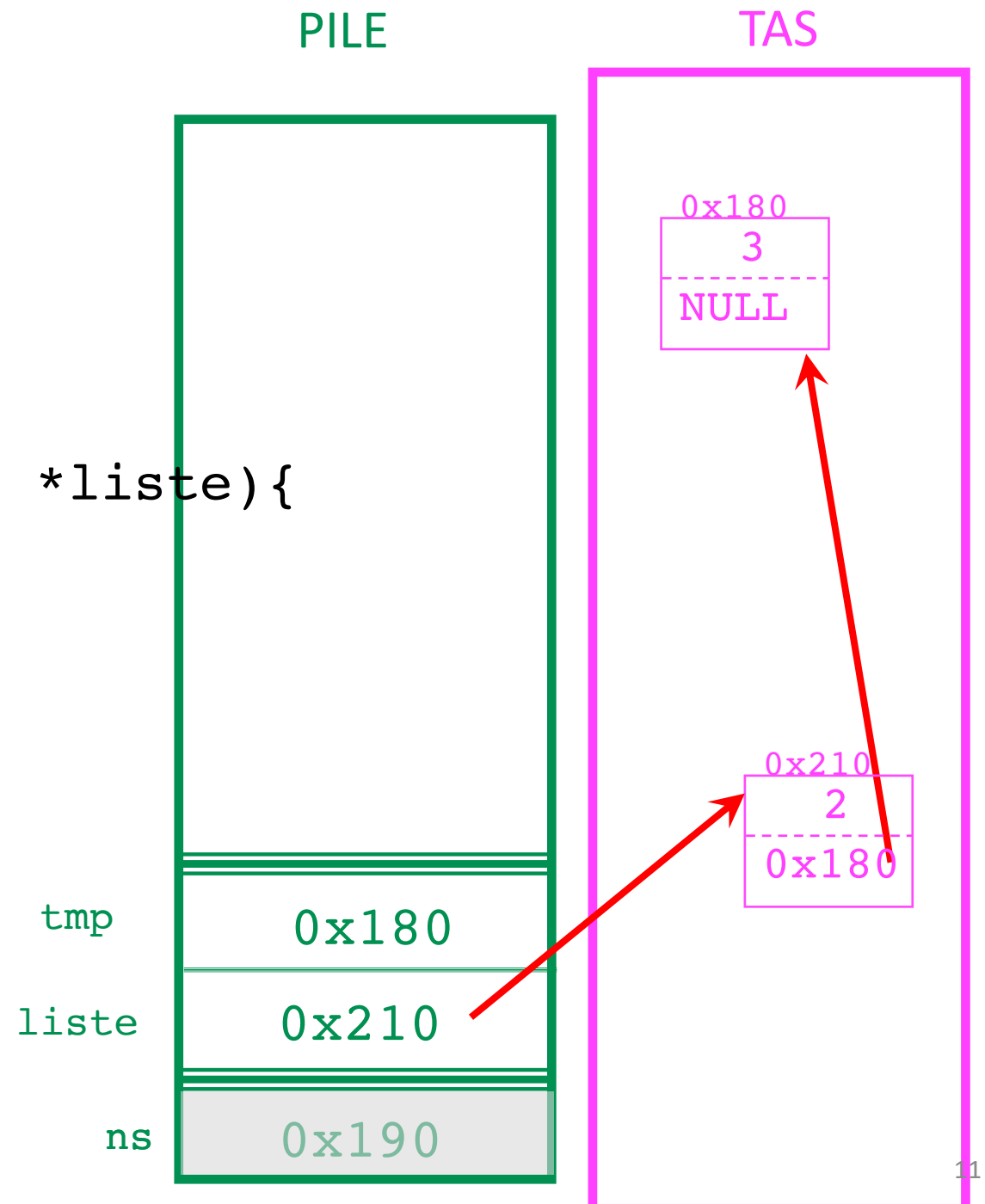
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
→       tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

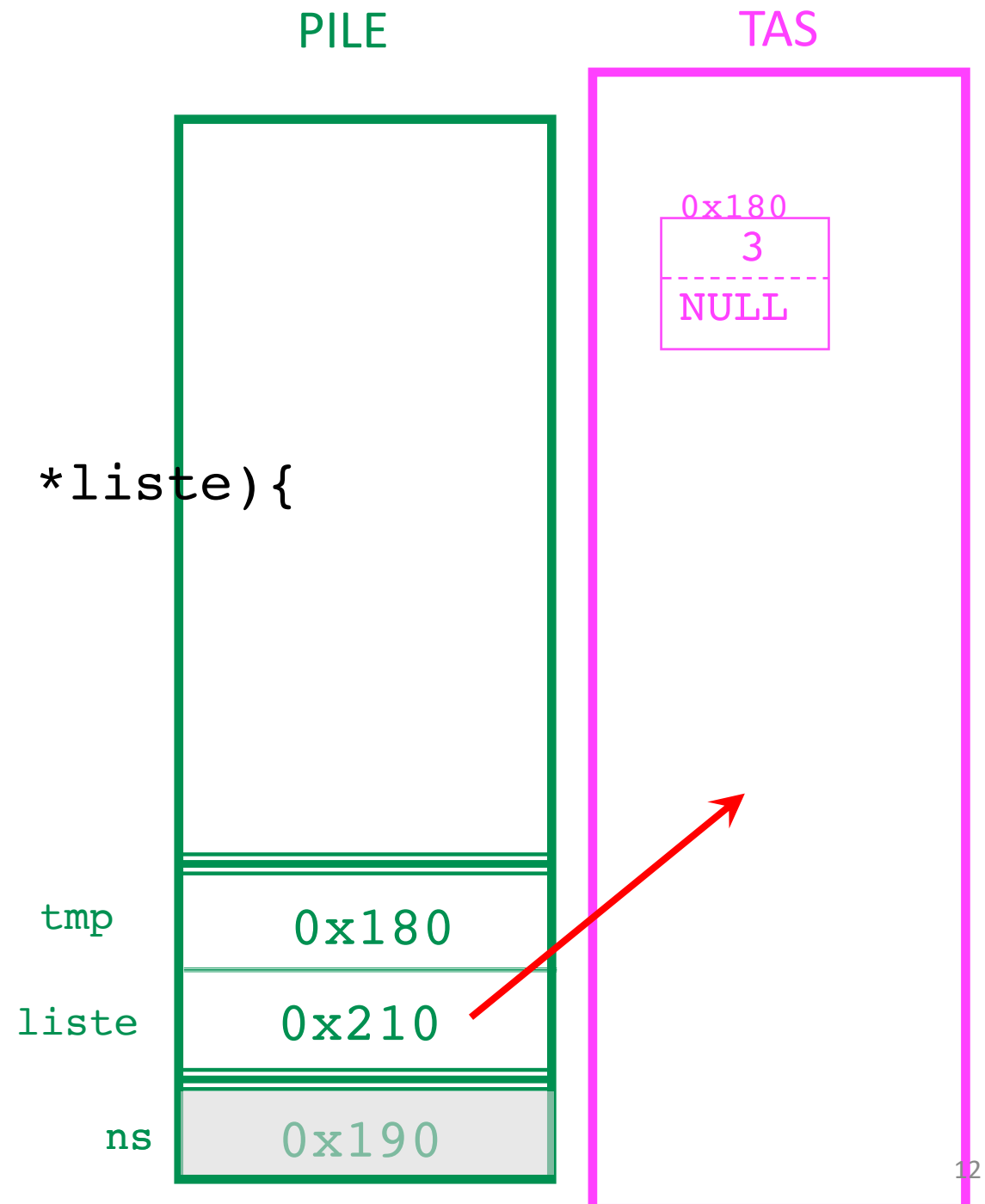
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        → free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

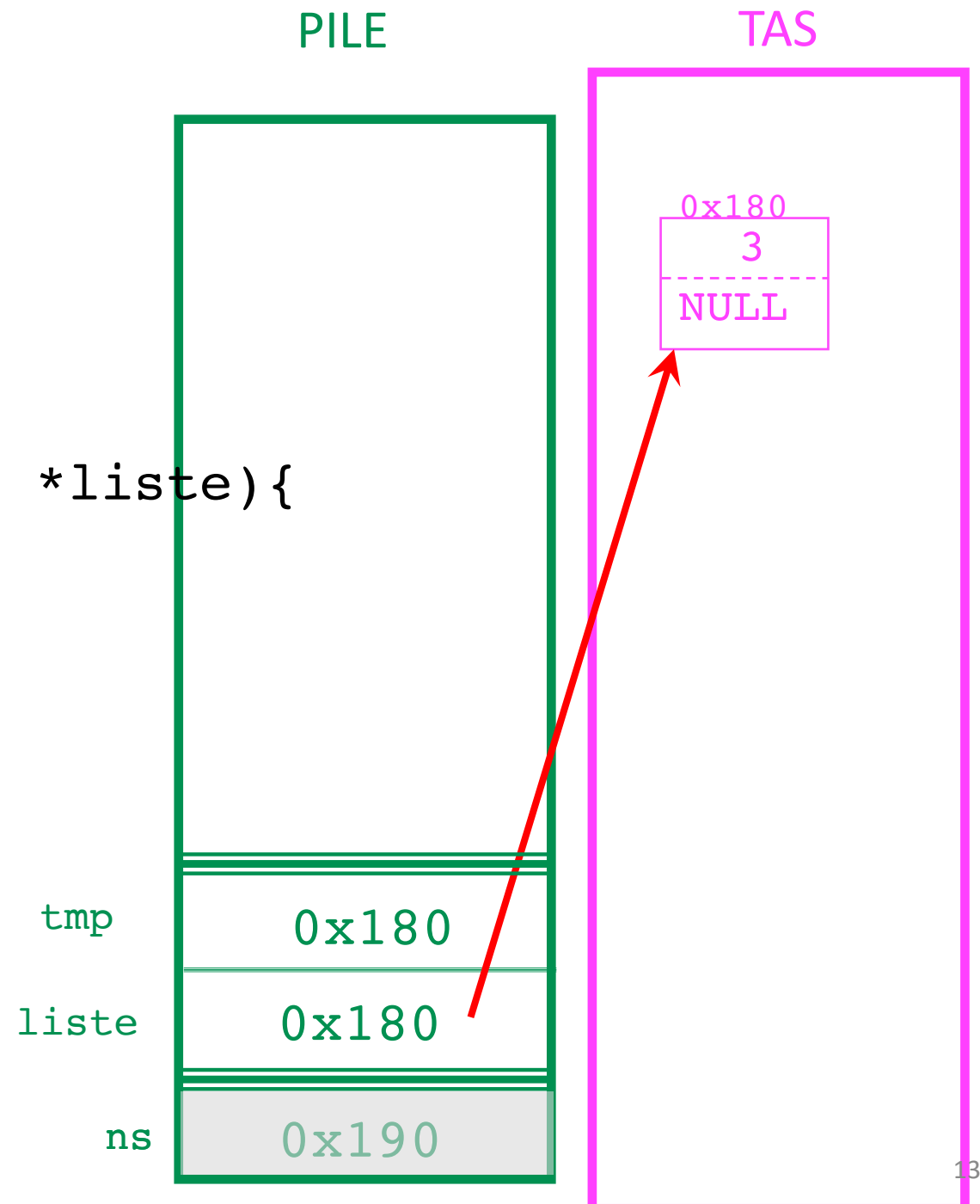
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

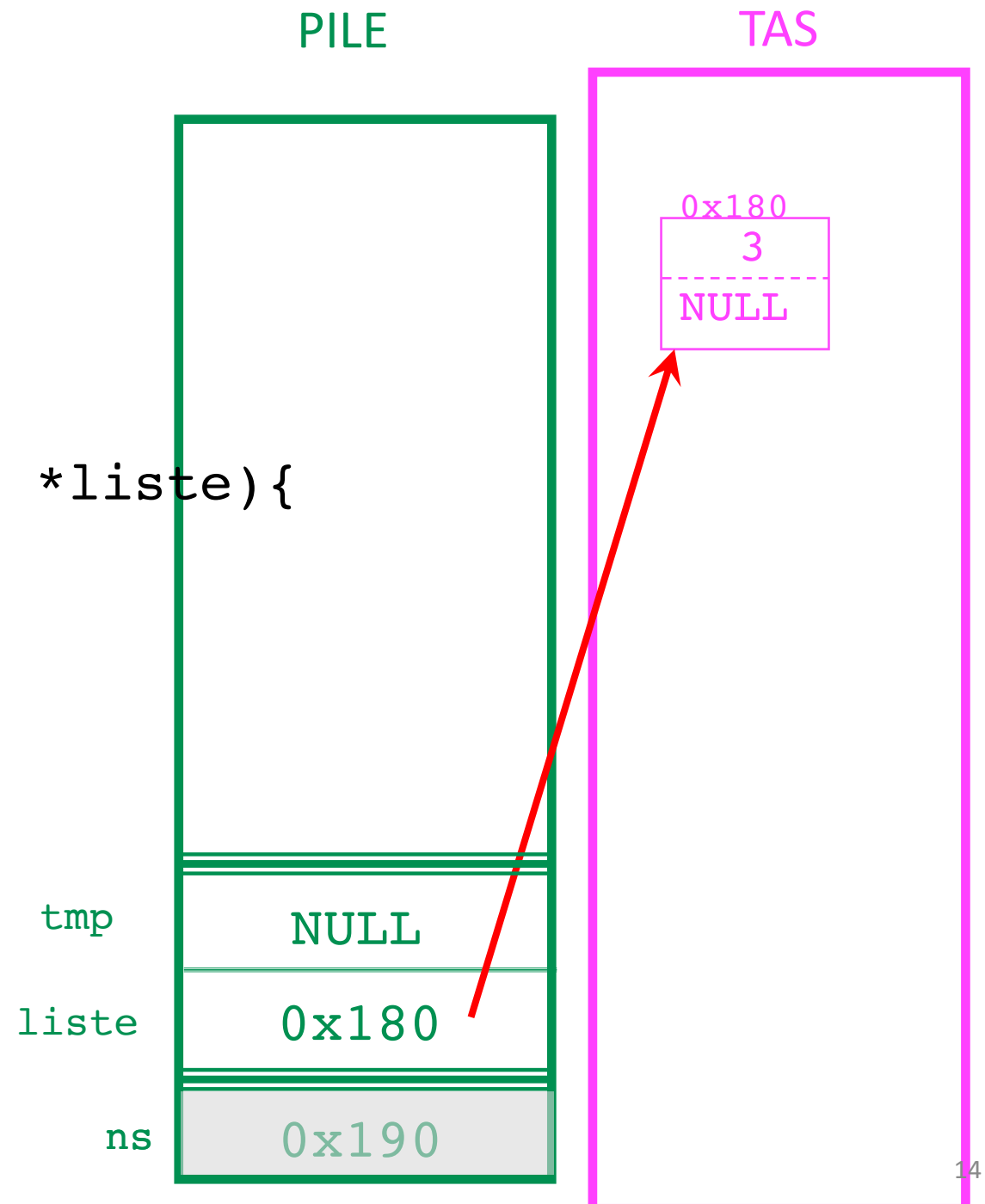
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
→       tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

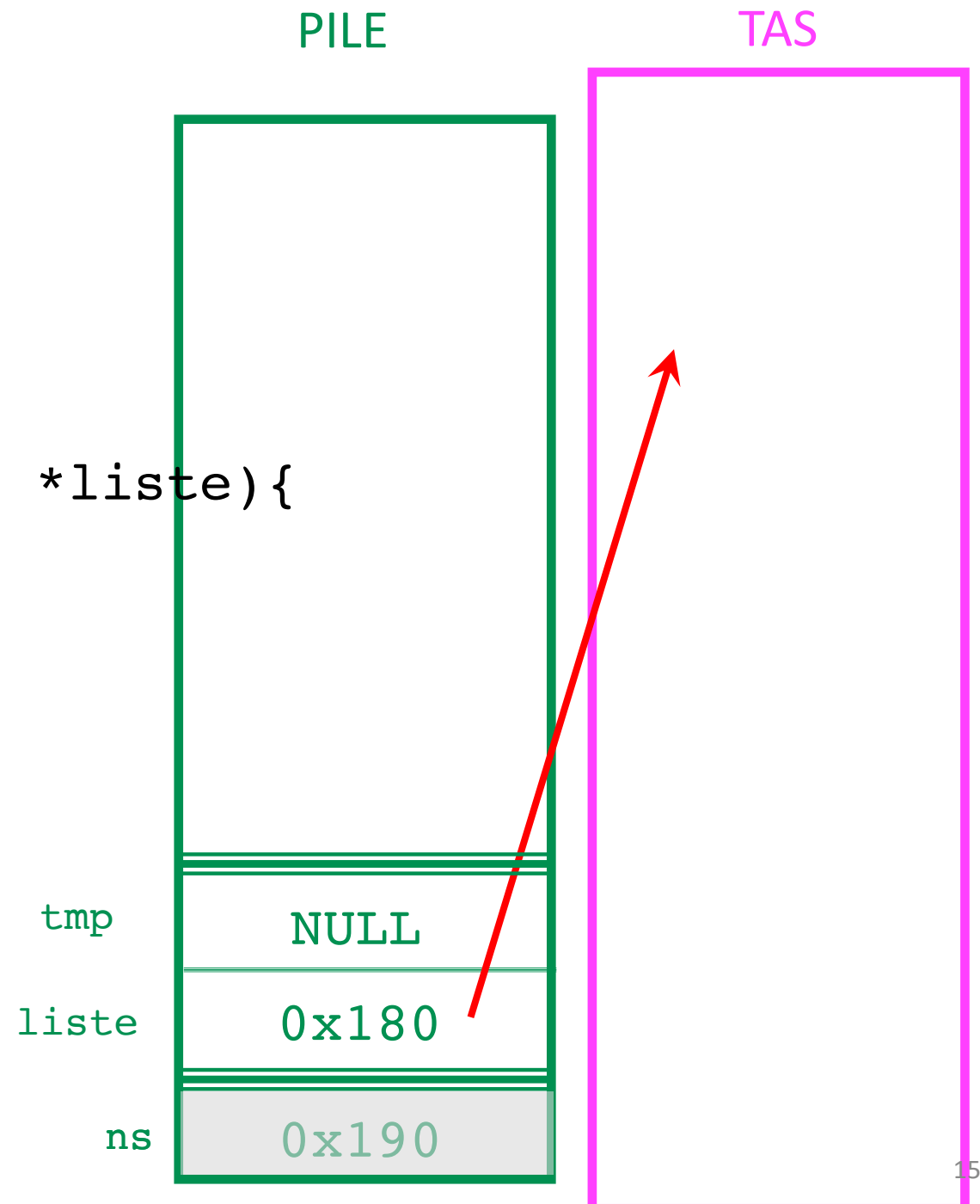
Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



Libérer une liste

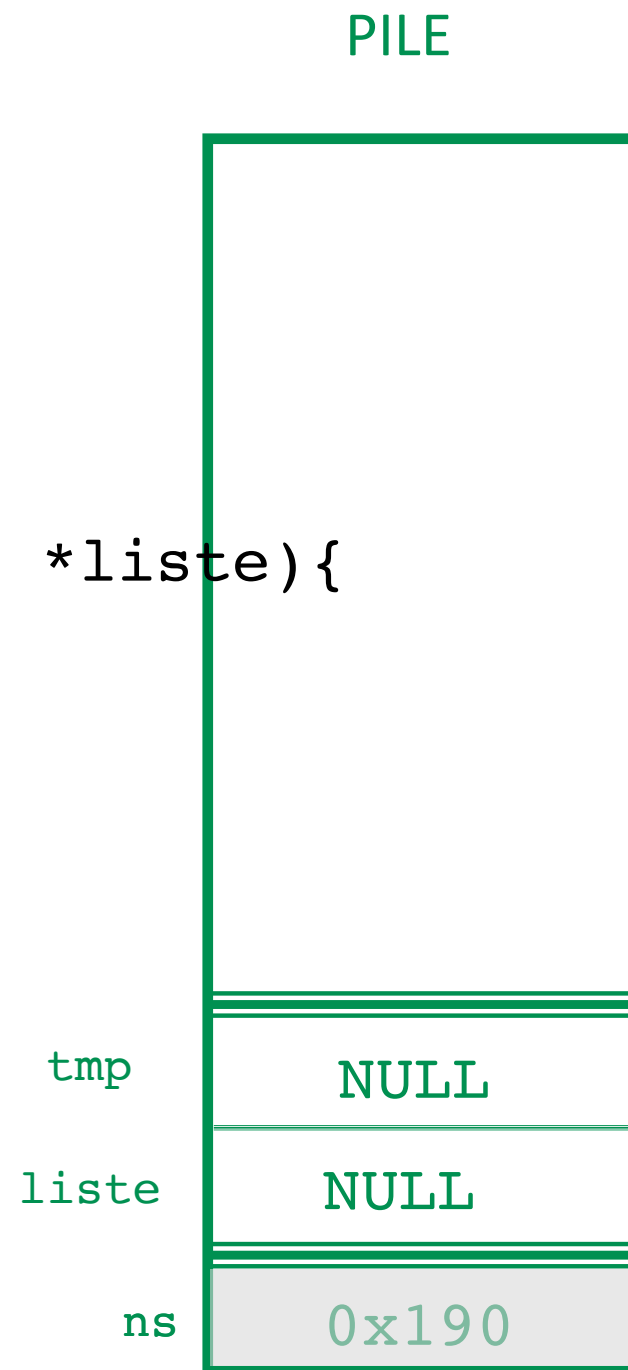
Comment faire ?

Pour chaque élément de la liste

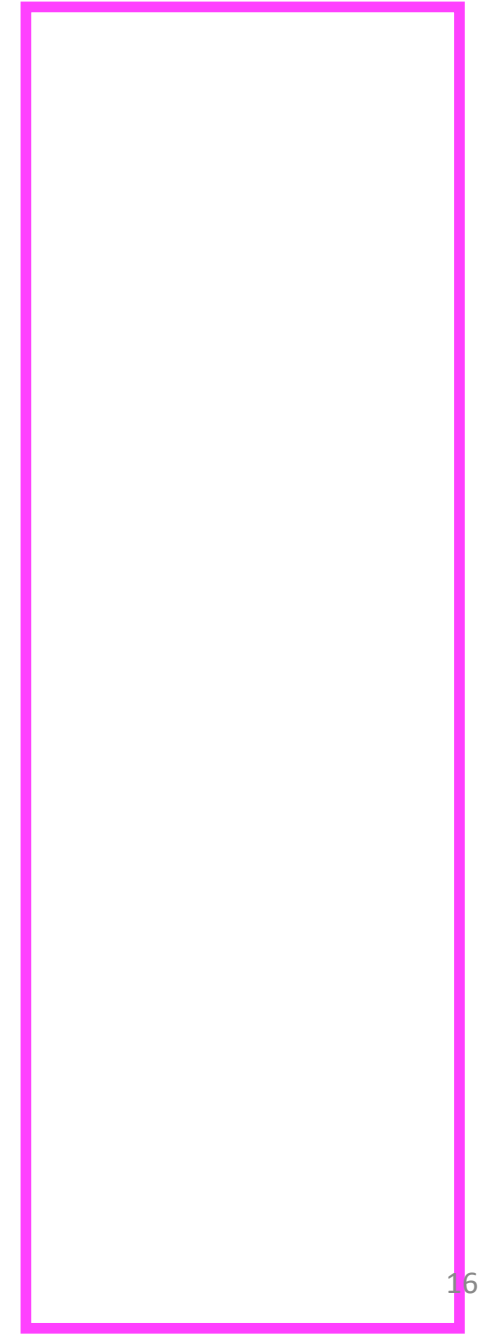
Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    return NULL;  
}
```



TAS



Libérer une liste

Comment faire ?

Pour chaque élément de la liste

Le libérer (free)

Passer au suivant

```
cellule_t *Liberer_liste(cellule_t *liste){  
    cellule_t *tmp;  
    while(liste != NULL){  
        tmp=liste->suivant;  
        free(liste);  
        liste=tmp;  
    }  
    → return NULL;  
}
```

PILE

TAS

tmp

NULL

liste

NULL

ns

0x190

Libérer une liste

```
cellule_t *Liberer_liste(cellule_t *liste){
    cellule_t *tmp;
    while(liste !=NULL){
        tmp=liste->suivant;
        free(liste);
        liste=tmp;
    }
    return NULL;
}
```

Jeu de test:

Cela marche t-il :

- pour une liste quelconque ?
- pour une liste vide ?

```
int main(){
    cellule_t *ns;
    ns=cons(1,cons(2,cons(3,NULL)));
    ns=Liberer_liste(ns);

    ns=Liberer_liste(ns);
    return 0;
}
```