Les tableaux unidimensionnels en C

Séance 7

de l'UE « introduction à la programmation »

Bruno Bouzy

bruno.bouzy@parisdescartes.fr

Tableaux unidimensionnels

- Déclaration

- Utilisation par valeur ou par pointeur

- Tableau de pointeurs
- Allocation dynamique
- Recopie rapide

Tableaux unidimensionnels

Déclaration:

```
int tab[5];
```

- `int` type des cases du tableau
- `tab` nom du tableau
- La taille du tableau est `5`
- Le type de `tab` est `int *`

Tableaux unidimensionnels

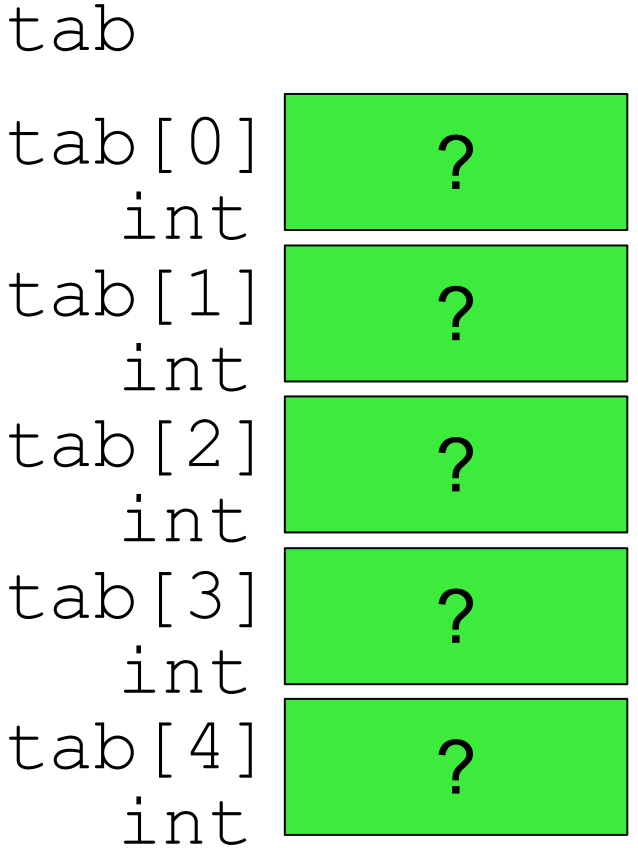
- Déclaration:

```
#define TAILLE 5
int tab[TAILLE];
```

  - `TAILLE` : taille du tableau

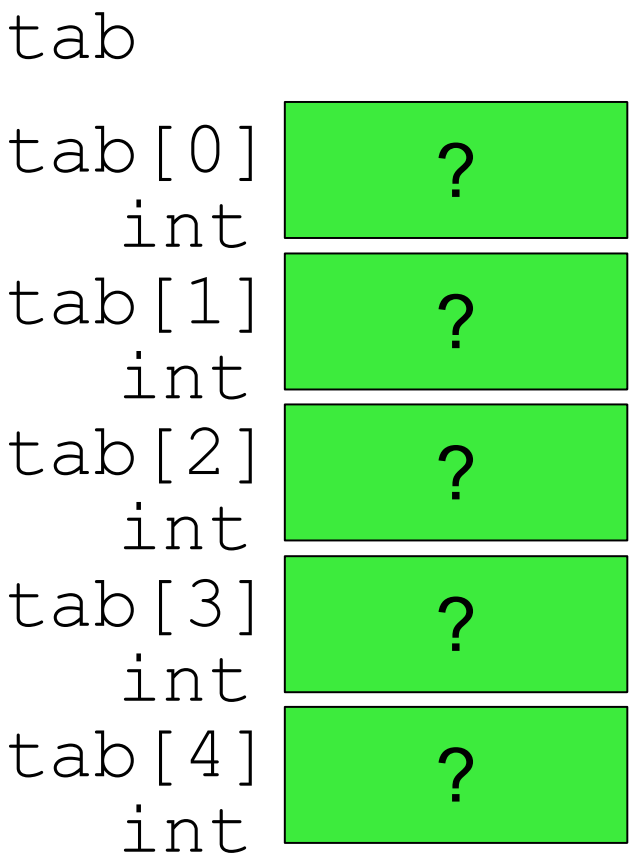
Exemple

- Emplacement mémoire
  - `5` cases mémoire
  - Type des cases : `int`
  - Numérotation de `0` à `4`
  - Initialisation des cases
    - `0` dans la plupart des cas
    - non garanti: `?`



Exemple

- Emplacement mémoire
  - `tab` : adresse de `tab[0]`
  - `tab = &tab[0]`

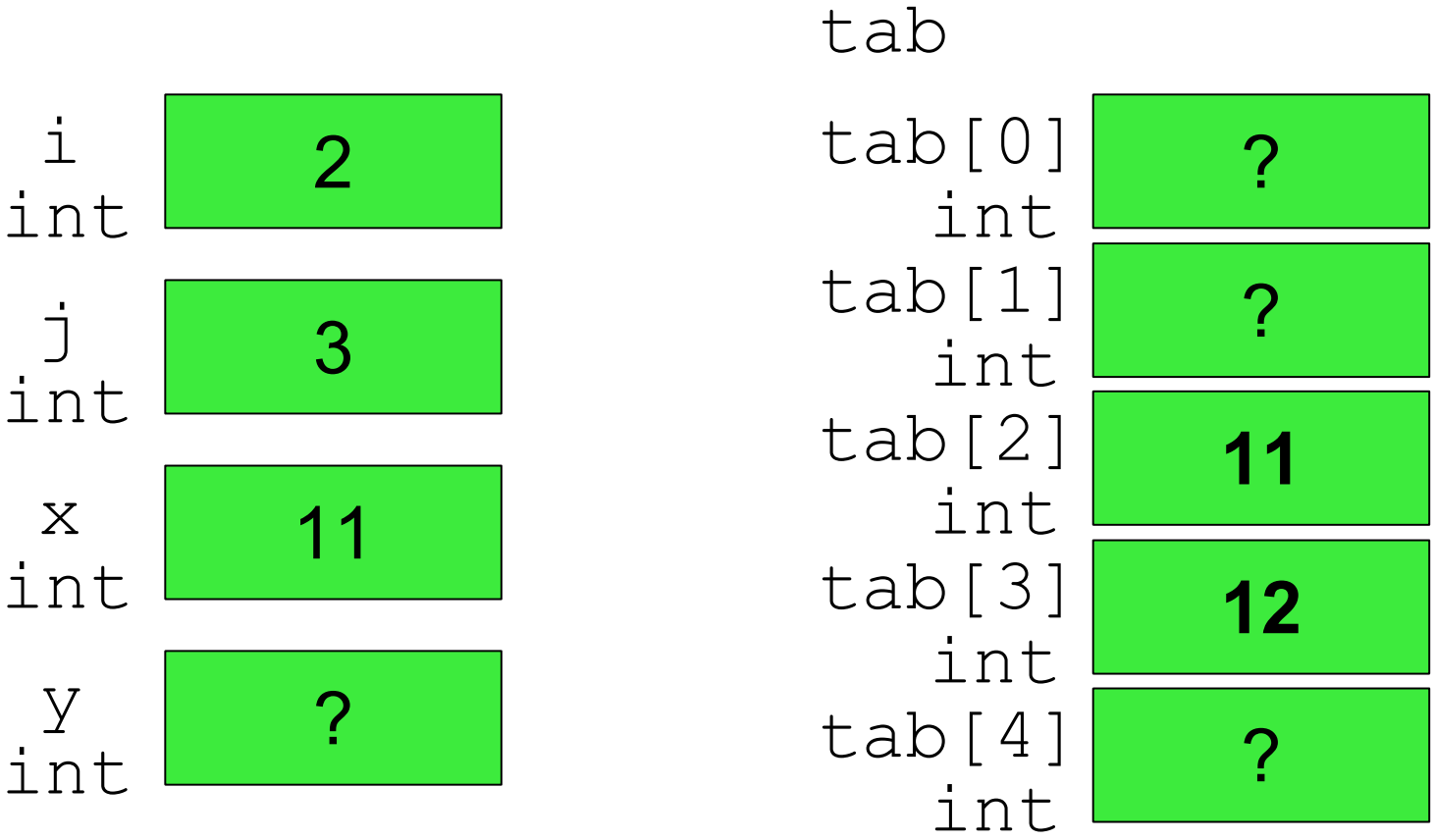


Tableaux unidimensionnels

- Utilisation par valeur:
  - Ecriture:
    - `int i=2, j=3, x = 11, y;`
    - `tab[i] = x; tab[j] = x+1;`

Exemple

- Lecture et écriture dans un tableau:

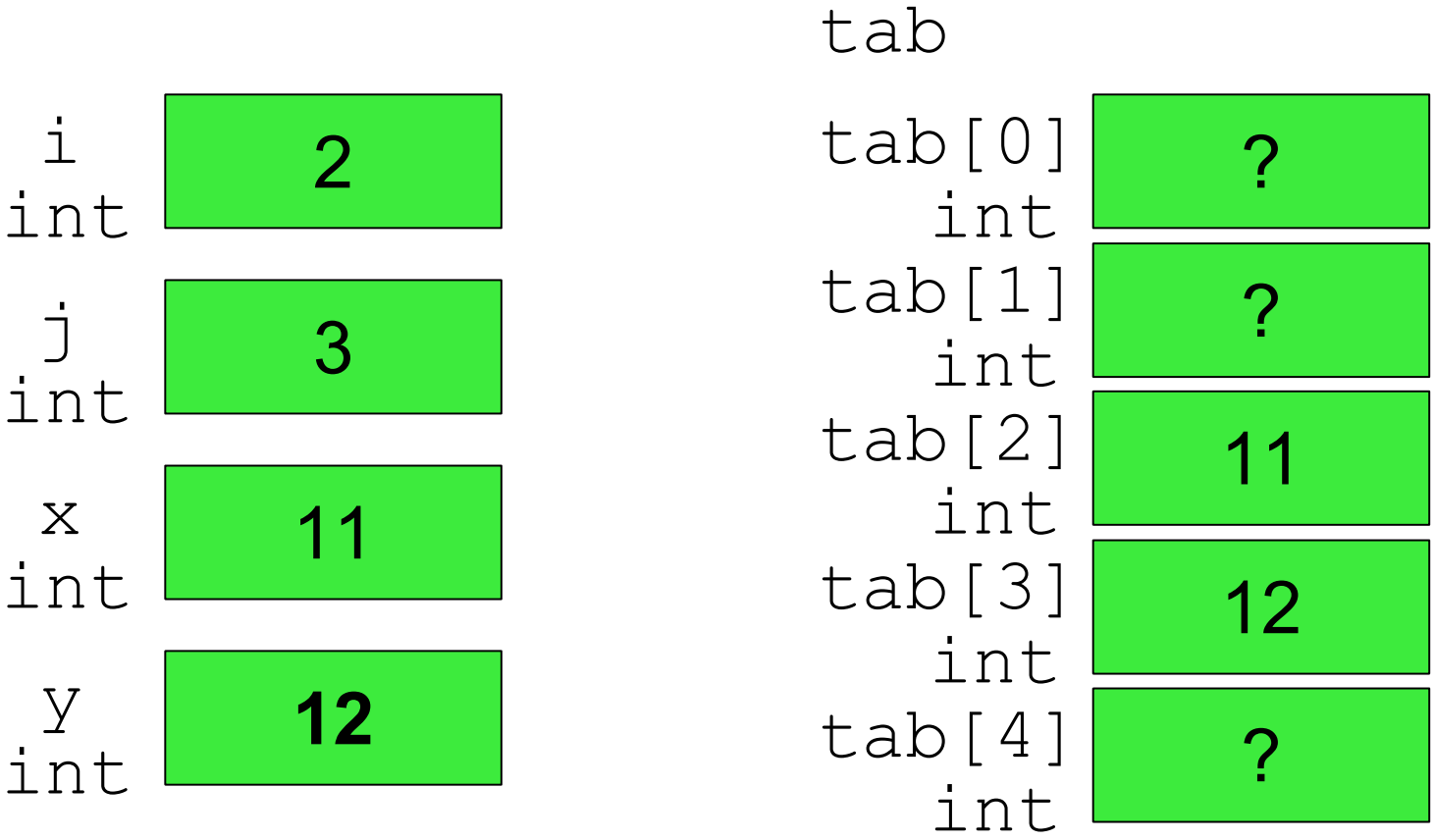


Tableaux unidimensionnels

- Utilisation par valeur:
  - Lecture:
    - `y = tab[j];`

Exemple

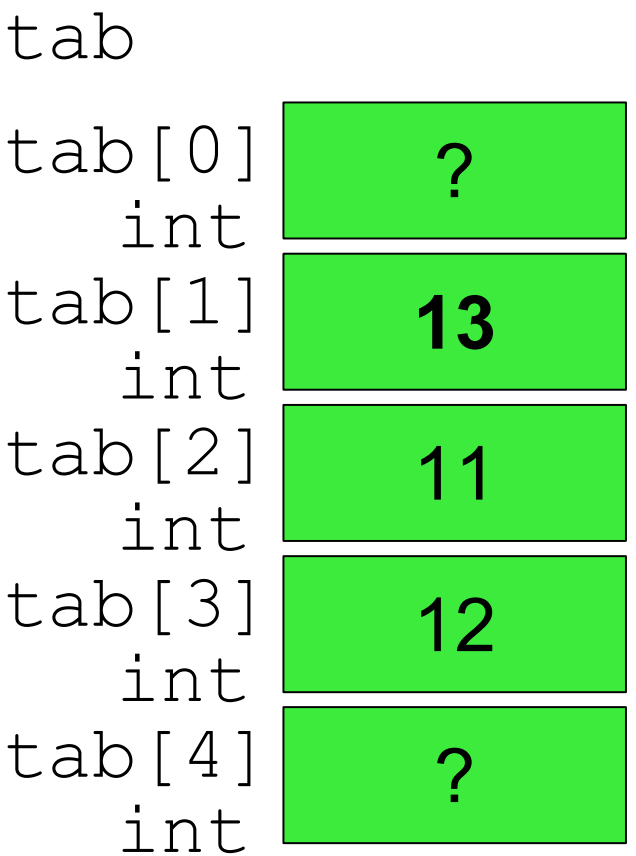
- Lecture et écriture dans un tableau:



Tableaux unidimensionnels

- Utilisation par index

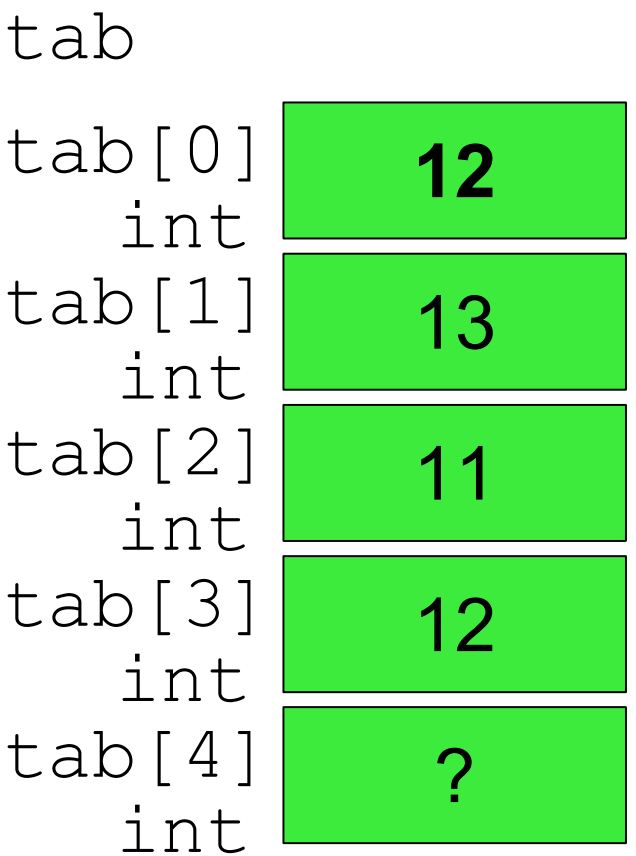
```
tab[1] = 13;
```



Tableaux unidimensionnels

- Utilisation par index

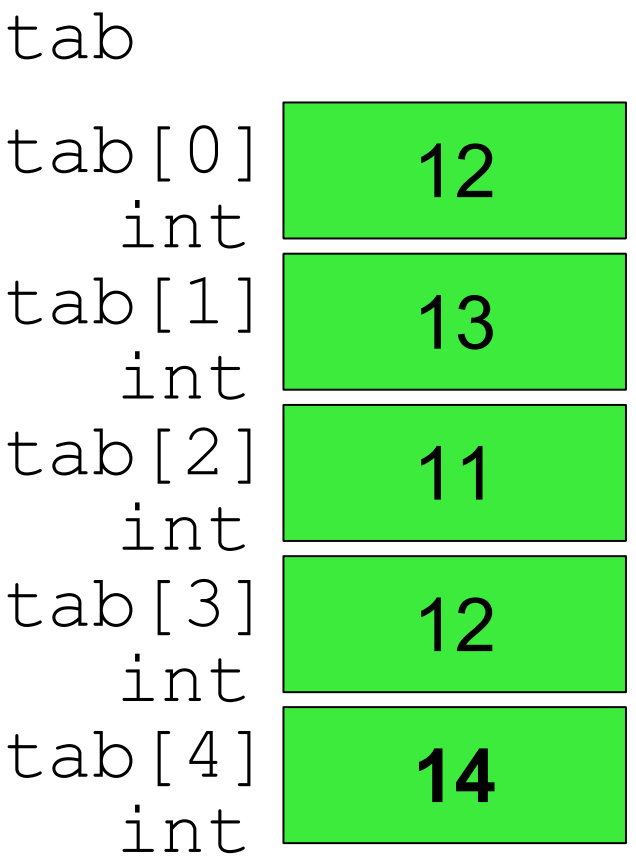
```
tab[0] = tab[3];
```



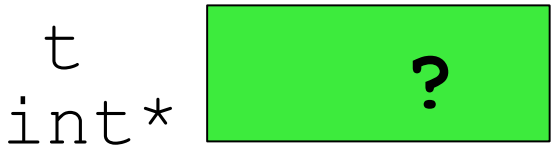
Tableaux unidimensionnels

- Utilisation par index

```
tab[4] = 14;
```

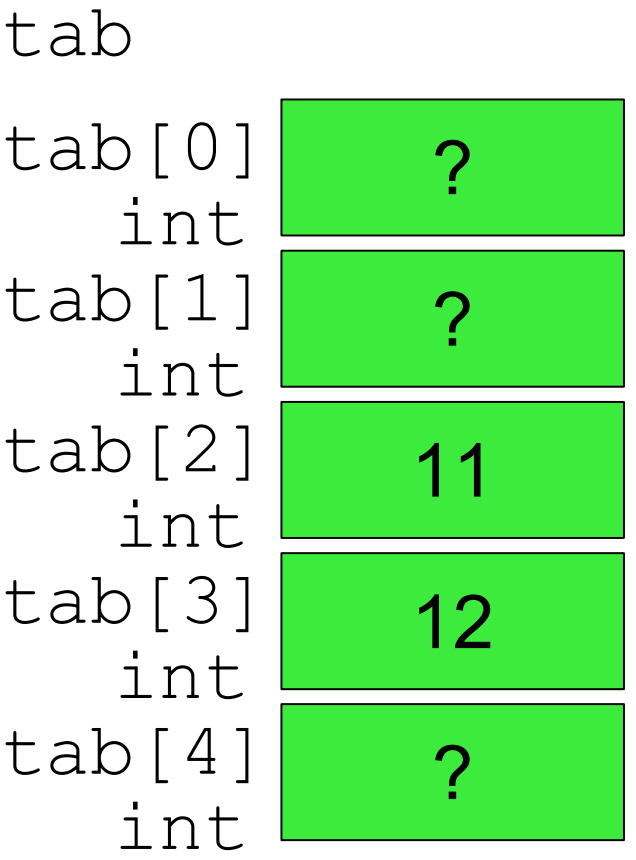


Tableaux unidimensionnels

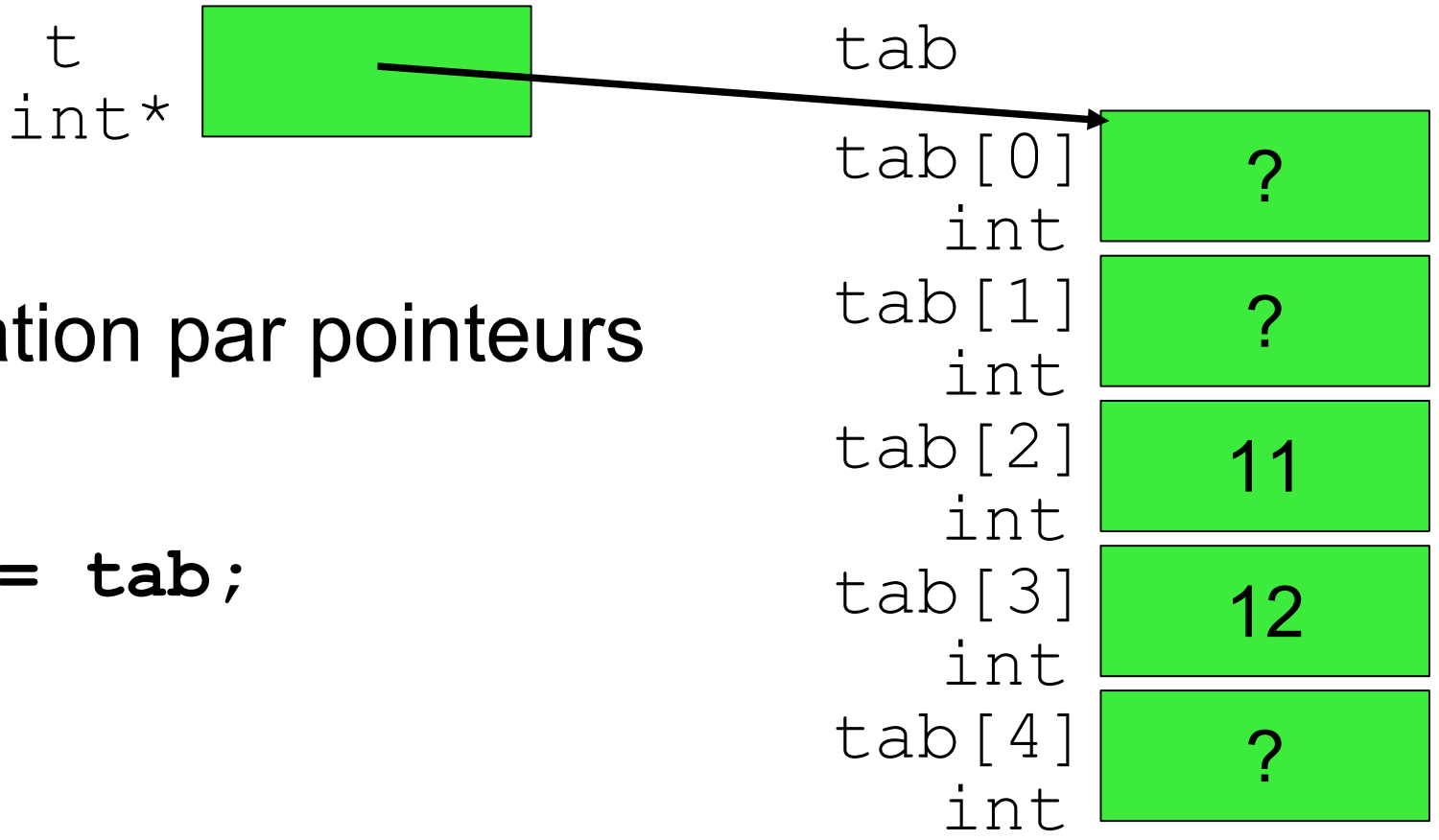


- Utilisation par pointeurs

```
int * t;
```



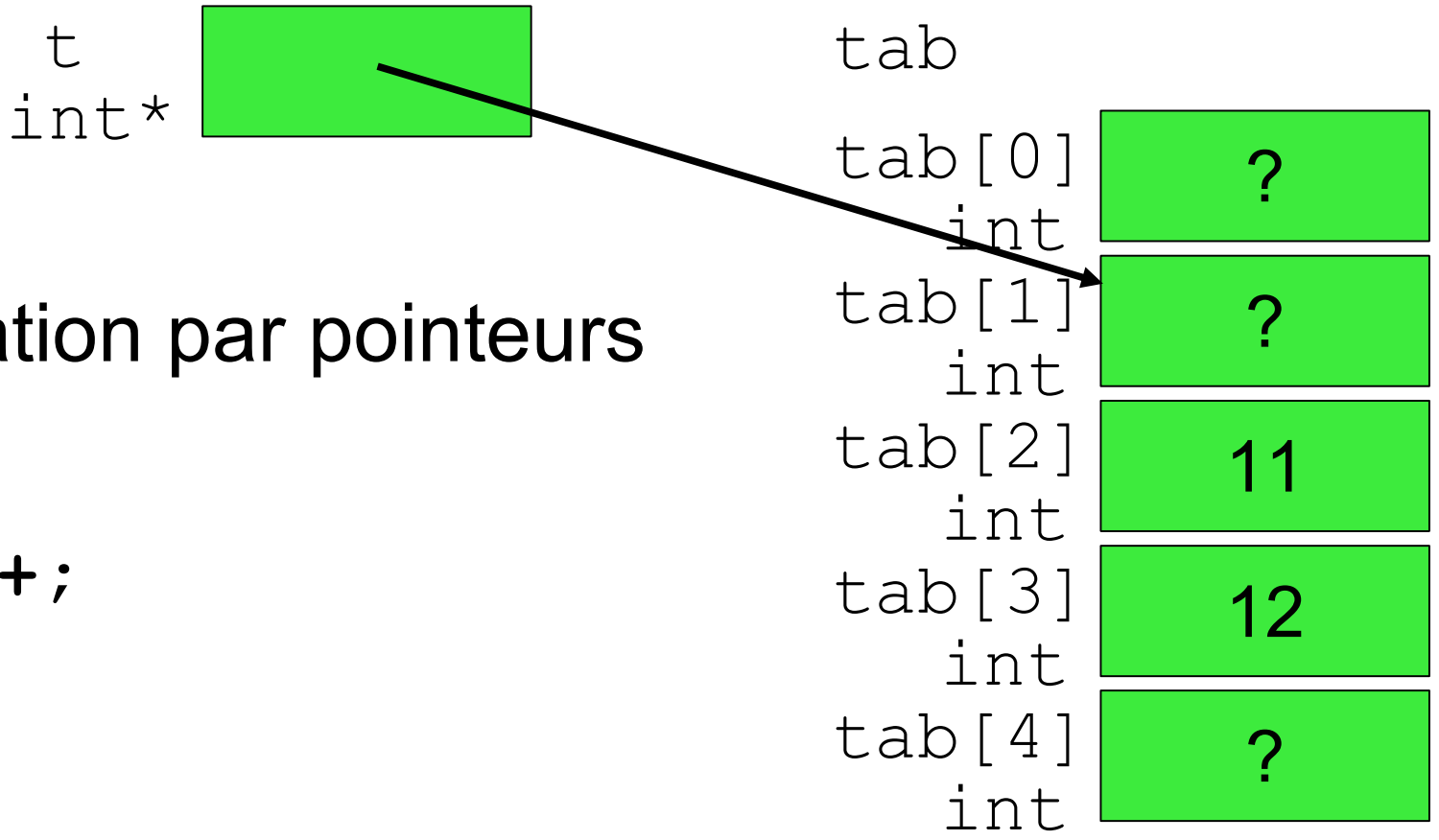
Tableaux unidimensionnels



- Utilisation par pointeurs

```
t = tab;
```

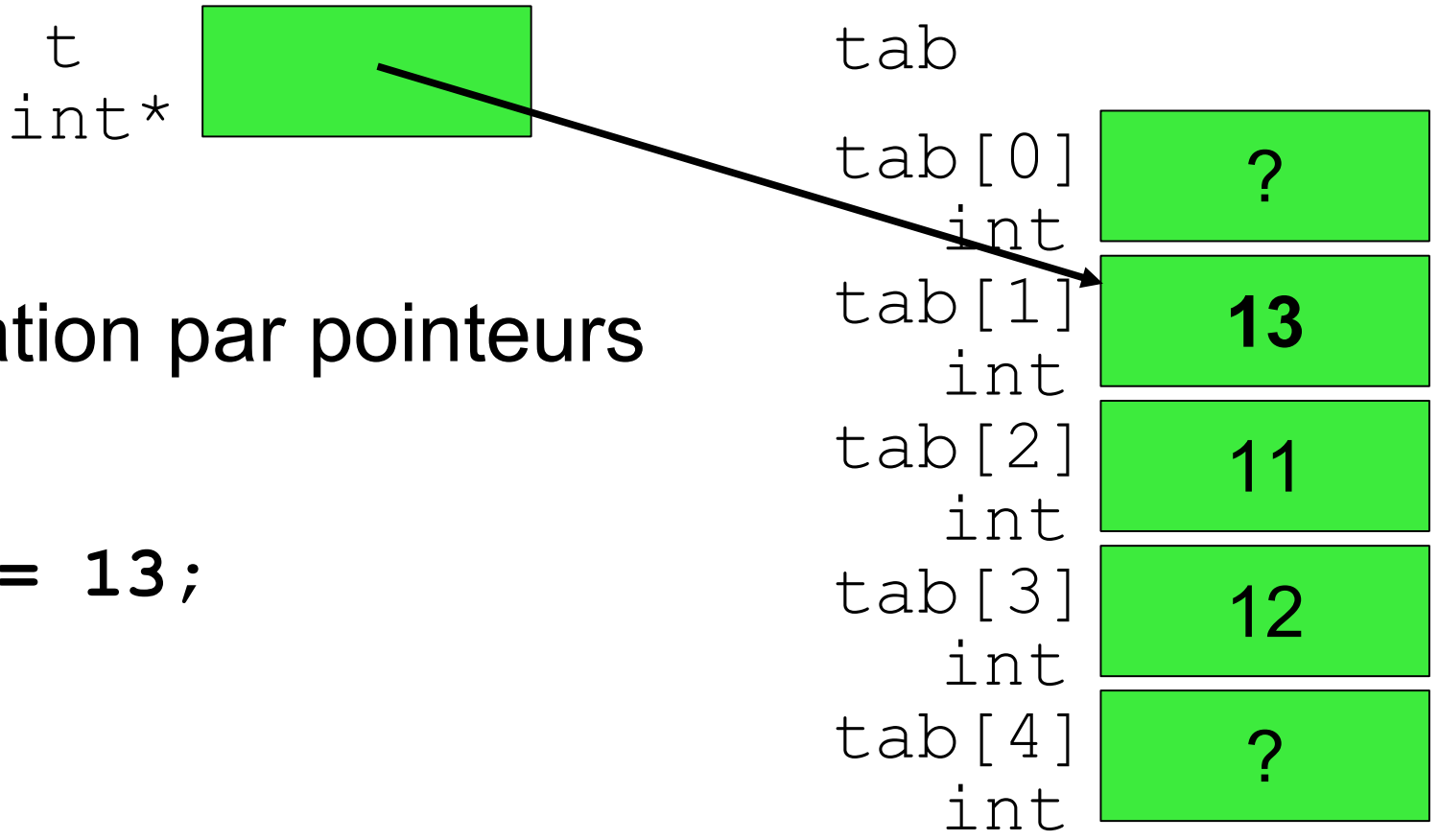
Tableaux unidimensionnels



- Utilisation par pointeurs

```
t++;
```

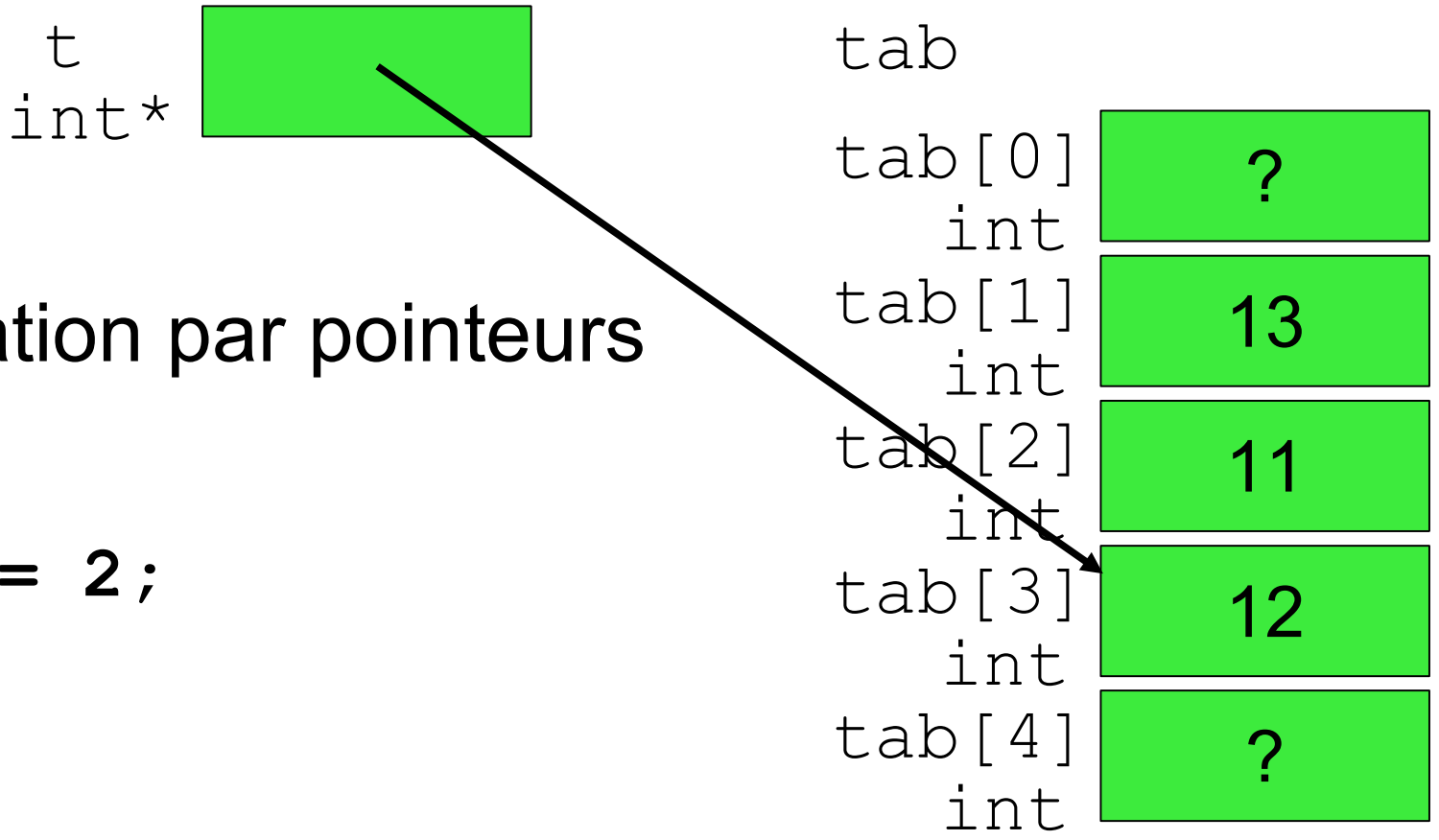
Tableaux unidimensionnels



- Utilisation par pointeurs

```
*t = 13;
```

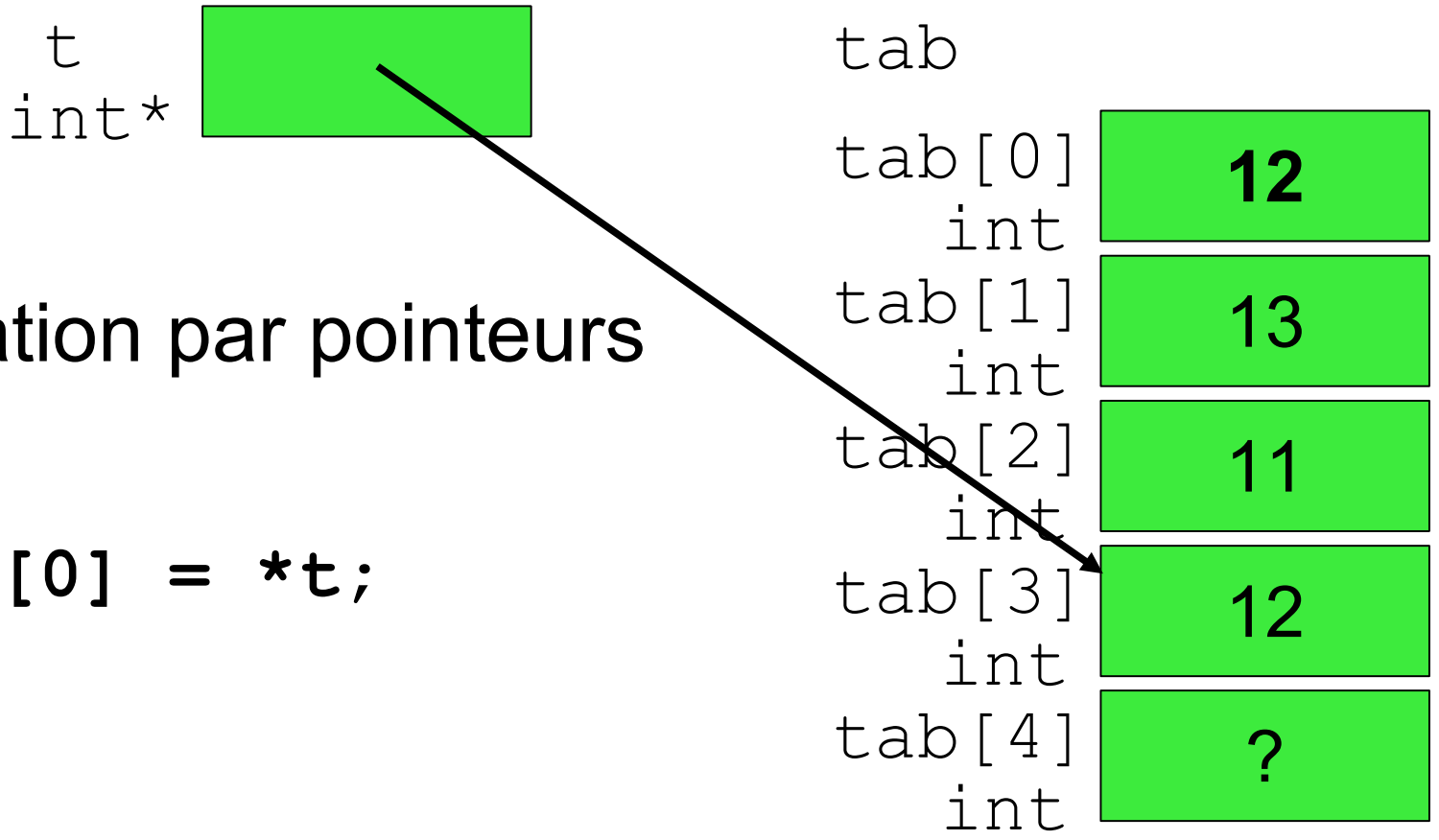
Tableaux unidimensionnels



- Utilisation par pointeurs

```
t += 2;
```

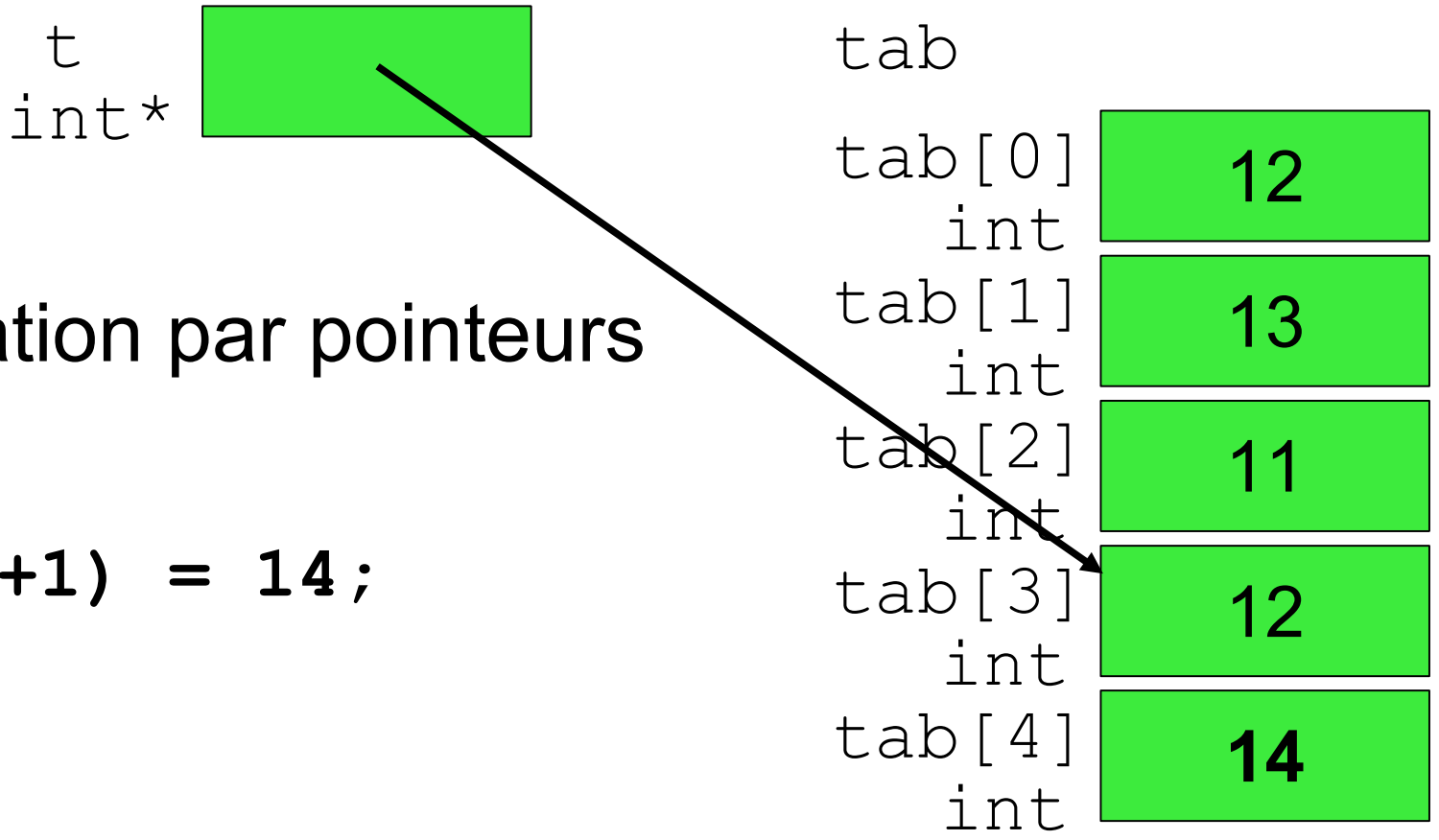
Tableaux unidimensionnels



- Utilisation par pointeurs

```
tab[0] = *t;
```

Tableaux unidimensionnels



- Utilisation par pointeurs

```
*(t+1) = 14;
```

Tableaux unidimensionnels

- Utilisation par pointeur:
  - `t+i` : adresse de la ième case suivant la case pointée par `t`
  - `t+i` équivaut à `&t[i]`
  - `*(t+i)` : contenu de la ième case suivant la case pointée par `t`
  - `*(t+i)` équivaut à `t[i]`

Exemple – formalisme index

```
#define NVAL 5
...

  int i, min, max;
  int t[NVAL];

  for (i=0; i<NVAL;i++) {
    printf("t[%d] ? ", i);
    scanf("%d", &t[i]);
  }

  max = min = t[0];
  for (i=1; i<NVAL;i++) {
    max = (t[i]>max) ? t[i] : max;
    min = (t[i]<min) ? t[i] : min;
  }

  printf("max = %d, min = %d.\n", max, min);
```

Exemple – formalisme pointeur

```
#define NVAL 5

...

  int *pt, i, min, max;
  int t[NVAL];

  for (pt=t, i=0; pt<t+NVAL; pt++, i++) {
    printf("t[%d] ? ", i);
    scanf("%d", pt);
  }

  max = min = *t;
  for (pt=t+1; pt<t+NVAL; pt++) {
    max = (*pt>max) ? *pt : max;
    min = (*pt<min) ? *pt : min;
  }

  printf("max = %d, min = %d.\n", max, min);
```

Résumé de la séance 7

- Déclaration
- Utilisation par valeur ou par pointeur

- Tableaux de pointeurs
- Allocation dynamique
- Recopie rapide