

# C/C++: Allocation dynamique.



Vincent Gaudillière

Polytech Nancy

2017 / 2018

## Allocation de mémoire.

---

Lorsqu'on déclare une variable, on demande en fait à l'OS de réserver l'espace nécessaire en mémoire pour stocker cette variable.

```
int a = 10; // sizeof(int) = 4 octets
```

```
double x; // sizeof(double) = 8 octets
```

```
char c; // sizeof(char) = 1 octet
```

```
float tab[15]; // 15*sizeof(float) = 15*4 = 60 octets
```

## Déclaration d'un tableau en C.

---

Comment déclarer un tableau en C?

Comment allouer la mémoire pour un tableau?

## Déclaration d'un tableau en C.

---

Comment déclarer un tableau en C?

Comment allouer la mémoire pour un tableau?

Deux possibilités: Si la taille du tableau...

- ... est connue avant la compilation  $\Rightarrow$  **Allocation statique**,
- ...n'est pas connue avant la compilation  $\Rightarrow$  **Allocation dynamique**.

## Allocation statique.

---

Déclaration de la taille d'un tableau:

```
int taille = 10;  
int mon_tableau[taille]; // INTERDIT (en C89) !
```

```
int mon_tableau[10]; // OK
```

```
#define HEIGHT_IMAGE 1080  
#define WIDTH_IMAGE 1920
```

```
unsigned char mon_image1[HEIGHT_IMAGE][WIDTH_IMAGE]; //OK  
unsigned char mon_image2[HEIGHT_IMAGE][WIDTH_IMAGE]; //OK
```

## Allocation dynamique.

---

```
int nbElements = 10; // taille stockée dans une variable!  
int * mon_tableau = NULL;  
  
mon_tableau = malloc(nbElements*sizeof(int)); // allocation  
  
if(mon_tableau!=NULL) // mémoire allouée  
{  
    mon_tableau[5] = 1; // opérations sur le tableau...  
    free(mon_tableau); // libérer la mémoire!  
}  
else  
{  
    printf("Erreur lors de l'allocation.\n");  
}
```

## Allocation dynamique: cas d'un tableau 2D.

---

```
int i;
int N = 10; // But: tableau de taille N*N
int ** mon_tableau = NULL;

mon_tableau = malloc(N*sizeof(int*));

for(i=0;i<N;i++){
    mon_tableau[i] = malloc(N*sizeof(int));
}

// mon_tableau:      type int **
// mon_tableau[i]:   type int *
// mon_tableau[i][j]: type int
```

## Allocation dynamique: cas d'un tableau 2D.

---

```
int i;
int N = 10;
int ** mon_tableau = NULL;

// Appels à malloc (cf slide précédent)...

// Libération de la mémoire:
for(i=0;i<N;i++){
    free(mon_tableau[i]);
}

free(mon_tableau);
```



## Annexe: Génération d'un nombre aléatoire.

---

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h> // ne pas oublier!

int main(){
    int n1,n2;

    srand(time(NULL)); // ne pas oublier!

    n1 = rand()%50; // entier aléatoire entre 0 et 49
    n2 = rand()%10; // entier aléatoire entre 0 et 9

    return 0;
}
```