

# Pointeurs de fonctions

Programmation séquentielle en C, 2021-2022

---

Orestis Malaspinas (A401), ISC, HEPIA

2021-11-16

Inspirés des slides de F. Glück

## Pointeurs de fonctions (1/3)

- Considérons la fonction `max` retournant la valeur maximale d'un tableau

```
int32_t max(int32_t *t, int32_t size) {
    int32_t val_max = t[0];
    for (int32_t i = 1; i < size; ++i) {
        if (t[i] > val_max) {
            val_max = t[i];
        }
    }
    return val_max;
}
```

- L'appel à `max`, retourne l'adresse de la fonction en mémoire.
- On peut affecter cette valeur à un pointeur.

## Pointeurs de fonctions (2/3)

- Le type de la fonction max est

```
int32_t (*pmax)(int32_t *, int32_t);
```

- Le type doit être déclaré avec la signature de la fonction.
- On peut alors utiliser l'un ou l'autre indifféremment

```
int32_t (*pmax)(int32_t *, int32_t);  
pmax = max;  
int32_t tab[] = {1, 4, -2, 12};  
printf("%d", max(tab, 4)); // retourne 12  
printf("%d", pmax(tab, 4)); // retourne 12 aussi
```

## Pointeurs de fonctions (3/3)

- On peut passer des fonctions en argument à d'autres fonctions

```
int32_t reduce(int32_t *tab, int32_t size, int32_t init,
              int32_t (*red_fun)(int32_t, int32_t))
{
    for (int32_t i = 0; i < size; ++i) {
        init = red_fun(init, tab[i]);
    }
    return init;
}
```

- Ici une fonction de *réduction* `sum()`

```
int32_t sum(int32_t lhs, int32_t rhs) {
    return lhs + rhs;
}

int32_t tab[] = {1, 4, -2, 12};
int32_t red = reduce(tab, 4, 0, sum);
printf("%d", red); // affiche 15
```