Cours programmation :les fonctions

Déclaration:

```
int mafonction(int);
```

Définition:

```
int mafonction(int x) {
int a; ...
return(a); }
```

Utilisation:

```
int x = 5; int y = mafonction(x);
```

Type de retour:

```
int mafonction(int);
```

Type du paramètre:

```
int mafonction(int);
```

Nom:

```
int mafonction(int);
```

Exemple 1: calcul du carré d'un nombre

```
int carre(int a) {                                  // définition
return (a*a);                                       // retour }
int main() {
 int n, x=0;
 printf("n ? ");
scanf("%d", &n); x = carre(n);                      // appel
printf("carre = %d\n", x);
return 0; }
```

Comment `carre` marche ?

- Après le `scanf`: n int 6 x int 0
- Appel de la fonction: a int 6
- Exécution du `return` return int 36
- Retour dans l'appelant n int 6 x int 36

Types de retour:

1. Les types pré-définis pour les variables:
   - int, float, double, char, etc.
   - int *, float *, char *, etc

2.Le type vide:
   - void

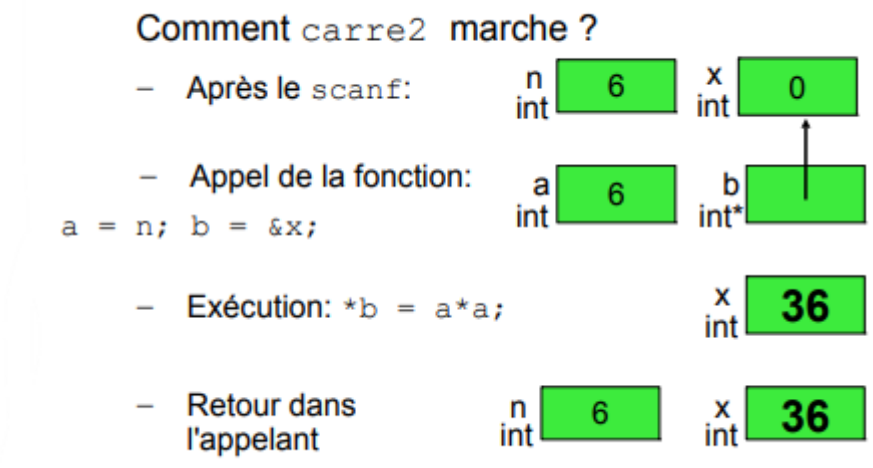
Paramètres en entrée ou sortie

1. Paramètre en entrée
   - La fonction ou la procédure en a besoin pour fonctionner

2.Paramètre en sortie
   - La fonction ou la procédure a pour objectif de lui donner une valeur.
   - Si le type du retour n'est pas void, une fonction C possède un paramètre de sortie obligé: le paramètre de retour.

Exemple 2 :carré avec passage de paramètres par **adresse**

```
void carre2(int a, int * b) {
*b = a*a;
}
int main() {
int n, x=0;
printf("n ? "); scanf("%d", &n);
 carre2(n, &x);
printf("carre2 = %d\n", x);
return 0;
}
```



Exemple 3 :carré avec passage de paramètres **par valeur**:

```
void carre3(int a, int b){
 b = a*a;
}
int main()
{
int n, x=0;
printf("n ? ");
scanf("%d", &n);
carre3(n, x);
printf("carre3 = %d\n", x);
return 0;
}
```

Comment `carre3` **marche pas** ?

- Après le `scanf`: n int 6 x int 0
- Appel de la fonction: a int 6 b int 0

`a = n; b = x;`

- Exécution `b = a*a;` b int **36**
- Retour dans l'appelant n int 6 x int **0**

Explication : b a été modifié correctement, pas x.