

Langage C

Fonctions et Procédures

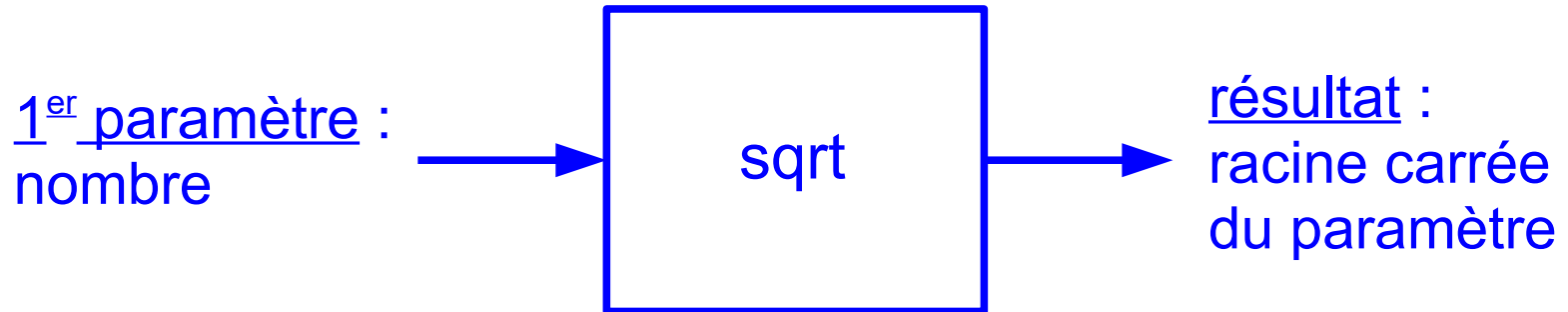
Calculer une racine carrée

```
#include <stdio.h>
#include <math.h>

int main ()
{
    float nb, resu;

    printf("Ton nombre : ");
    scanf("%f", &nb);
    resu = sqrt(nb);
    printf ("La racine carree de %f est %f\n", nb, resu);
    return 0;
}
```

sqrt



```
float nb, resu;  
  
nb = 2.0;  
resu = sqrt(nb);  
printf("Resu = %f\n", resu);
```

```
Resu = 1.414214
```

Utiliser une fonction

- ◆ Fournir le nombre de paramètres attendu par la fonction



```
sqrt(2)
```



```
sqrt(2, 25)
```

- ◆ Fournir des paramètres ayant le type attendu par la fonction



```
sqrt(25)
```



```
sqrt("vingt-cinq")
```

- ◆ Récupérer le résultat de la fonction



```
float racine ;  
racine = sqrt(25);
```



```
sqrt(25);
```

Prototype sqrt

◆ Racine carrée

```
double sqrt(double x);
```

résultat :
nombre à virgule

1^{er} paramètre :
nombre à virgule

```
float nb, resu;
```

```
nb = 2.0;
```

```
resu = sqrt(nb + 1.0);
```

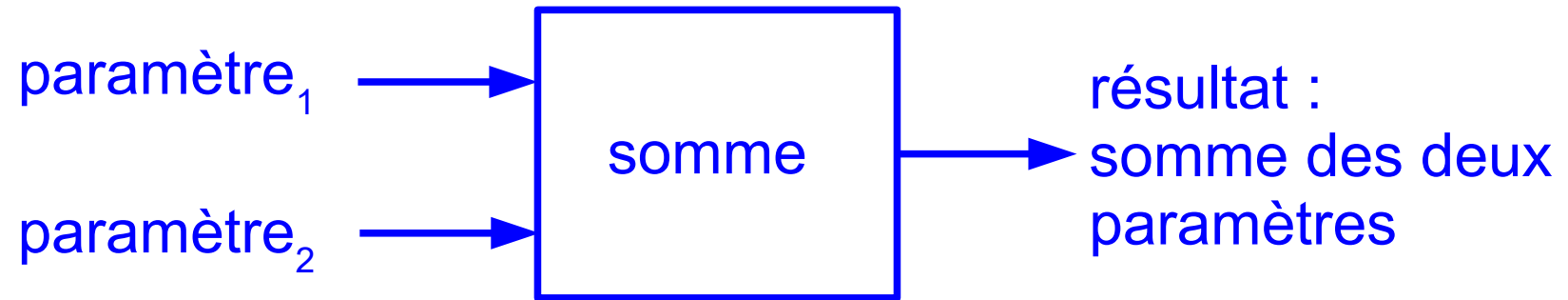
```
printf ("Resu = %f\n", resu);
```

```
Resu = 1.732051
```

Créer une fonction

- ◆ Objectif de la fonction ...
- ◆ ... définir cette fonction ...
- ◆ ... utiliser cette fonction

Somme de deux nombres entiers



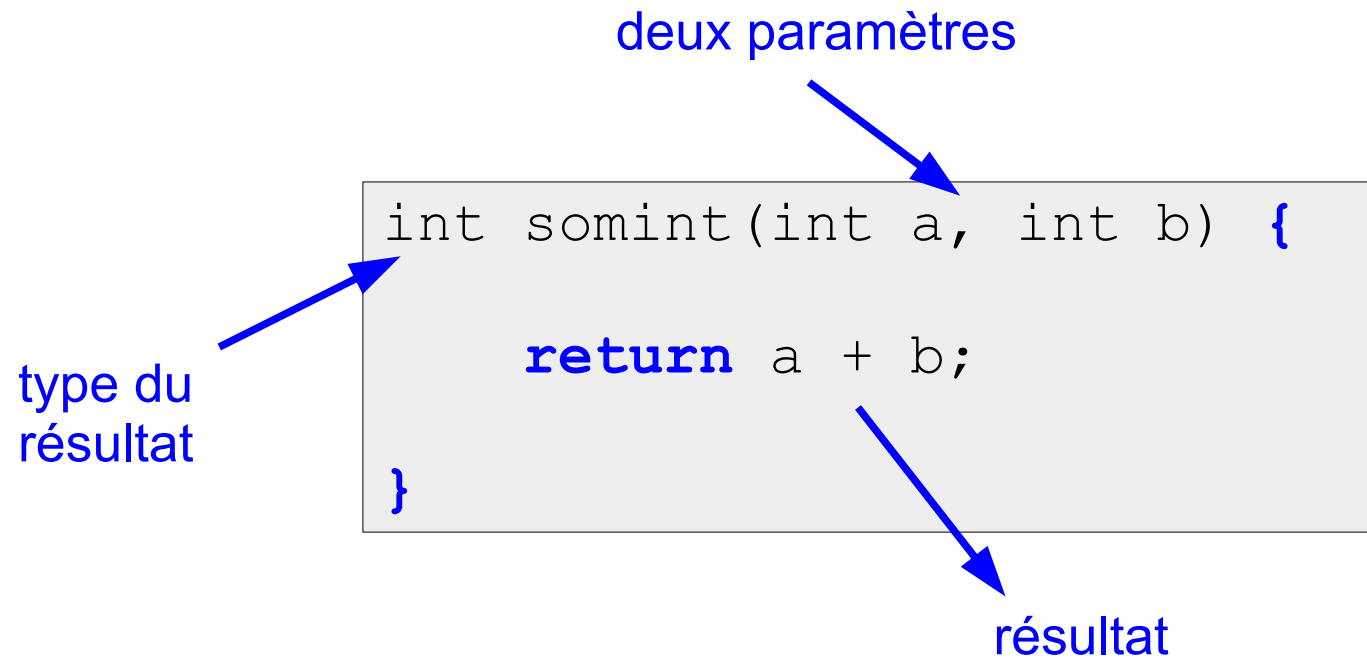
Prototype de « somint »

```
int somint(int a, int b);
```

résultat :
nombre entier

deux paramètres :
deux nombres entier

Définition de « somint »



Utilisation de « somint »

```
#include <stdio.h>
```

```
int somint(int a, int b) {  
    return a + b;  
}
```

définition

```
int main () {  
    int calcul;
```

```
    calcul = somint(5, 4);  
    printf("%d", calcul);  
    return 0;
```

utilisation

```
}
```

Exécution de « somint »

```
#include <stdio.h>

int somint(int a, int b) {
    return a + b;
}

int main () {
    int calcul;

    calcul = somint(5, 4);
    printf("%d", calcul);
    return 0;
}
```

1

Calcul de la valeur
des 2 paramètres :

1^{er} paramètre vaut 5

2^{ème} argument vaut 4

2

Affectation de ces valeurs
aux 2 arguments :

a prend la valeur 5

b prend la valeur 4



Passage par
valeur

3

Exécution des instructions
de la fonction « somint »

4

Calcul du résultat de
la fonction « somint »

La fonction retourne 9

Et le prototype ?

```
#include <stdio.h>

int somint(int a, int b) {
    return a + b;
}

int main () {
    int calcul;

    calcul = somint(5, 4);
    printf("%d", calcul);
    return 0;
}
```

```
#include <stdio.h>

int somint(int a, int b);

int main () {
    int calcul;

    calcul = somint(5, 4);
    printf("%d", calcul);
    return 0;
}

int somint(int a, int b) {
    return a + b;
}
```

Je rigole

```
#include <stdio.h>
```

```
int main () {  
    int cpt;
```

1^{ère} ligne



```
    for (cpt = 1; cpt <= 5; cpt++) {  
        printf("HA ");  
    }
```

2^{ème} ligne

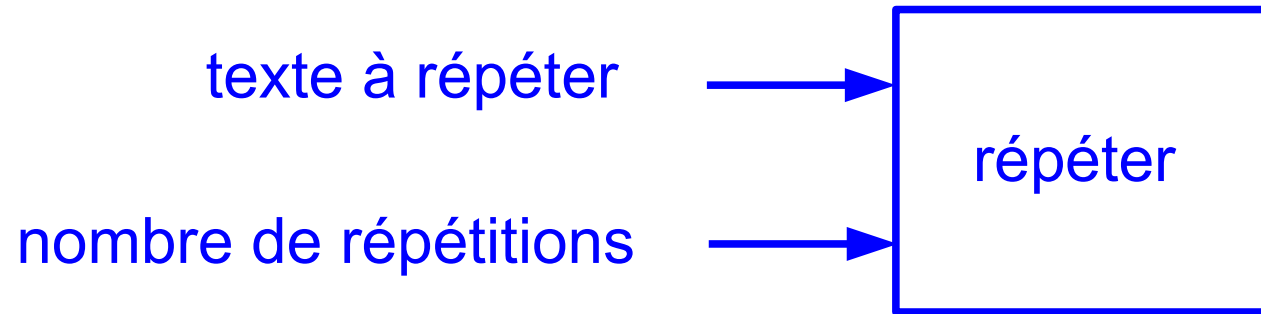


```
    printf("\n");  
    for (cpt = 1; cpt <= 3; cpt++) {  
        printf("Hi ");  
    }  
    printf("\n");  
    return 0;
```

```
}
```

```
HA HA HA HA HA  
Hi Hi Hi
```

Afficher plusieurs fois le même texte



- ◆ effet :
affichage du même texte
suivi d'un retour à la ligne
- ◆ résultat :
aucune valeur

Définir une procédure

Absence de
résultat



1^{er} paramètre



2^{eme} paramètre



```
void repeter(int combien, char texte[]) {  
    int cpt;  
    for (cpt = 1; cpt <= combien; cpt++) {  
        printf("%s", texte);  
    }  
    printf("\n");  
}
```

Syntaxe

Absence de
résultat



1^{er} paramètre



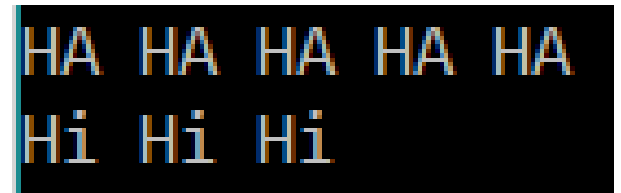
```
void nom_procedure (type nom_param , ... ) {  
    /* instructions de la procédure */  
}
```


Utiliser une procédure

```
#include <stdio.h>
```

```
void repeter(int combien, char texte[]) {  
    int cpt;  
    for (cpt = 1; cpt <= combien; cpt++) {  
        printf("%s", texte);  
    }  
    printf("\n");  
}
```

```
int main () {  
1ère ligne ← repeter(5, "HA ");  
2ème ligne ← repeter(3, "Hi ");  
    return 0;  
}
```



```
HA HA HA HA HA  
Hi Hi Hi
```

Prototype de « repeter »

```
void repeter(int combien, char texte[]);
```

résultat :
aucun

deux paramètres :
un nombre entier
une chaîne de caractères

En langage C, tout est fonction ...

- ◆ Une procédure est une fonction de type void