

INFO-H-100

Séances d'exercices 5 et 6

Les fonctions

Fonction

- Permet de découper un programme en modules fonctionnels.
- Une fonction est un ensemble d'instructions réalisant une certaine tâche.
- Une fonction
 - calcule une valeur (sinus, valeur absolue, racine carrée, ...)
 - exécute une tâche fréquente (imprimer un résultat formaté, imprimer un tableau, ...)

Paramètres et valeur de retour

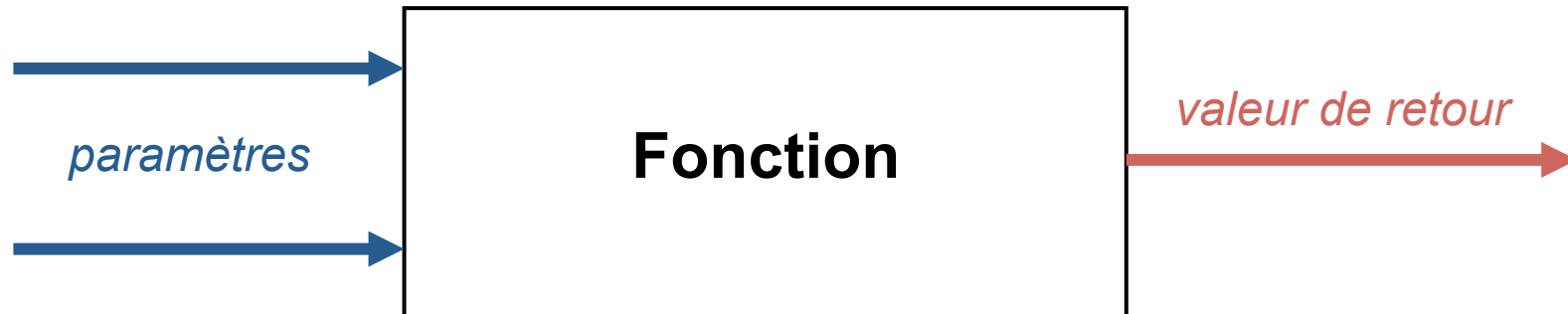
- Une fonction « prend » (ou non) un ou plusieurs *paramètres*.

Ex : sinus *d'un angle*, somme *de deux réels*.

- Une fonction « renvoie » (ou non) **UNE** *valeur de retour*.

Ex : La somme de 7.14 et de 3.1 est *10.24*.

Ex : Une fonction *printStars* qui reçoit 3 et affiche 3 étoiles (ne renvoie rien)



Déclaration d'une fonction

- `Type nom (paramètres)`
- Type de valeur de retour (`int`, `void`, `double`, `bool`, ...).
- `Nom (cos, abs, estPair, ...)`.
 - Par convention, le nom d'une fonction commence par une minuscule.
- Types et noms des paramètres. (`int i`, `double angle`, `double result`, `char c`, ...).
- Exemples de déclarations :
 - `int abs(int i);`
 - `double cos(double angle);`
 - `bool estPair(int n);`
 - `double somme(double i, double j);`
 - `void print(double result);`
- Déclaration ~ prototype ~ signature.
- Il faut déclarer une fonction avant de l'utiliser.

Définition d'une fonction

```
int abs(int i)
{
    int result;
    if(i < 0)
        result = -i;
    else
        result = i;
    return result;
}
```



return « renvoie »
une valeur compatible
avec le type de retour
de la fonction **et arrête**
la fonction.

Bonne pratique : un seul
return par fonction, à
la dernière ligne.

Exemples complets

```
#include <iostream>
using namespace std;

int abs(int i);

int main()
{
    int a;
    cin >> a;
    cout << abs(a) << endl;
    cout << abs(a-5) << endl;
    a = 2+(abs(a*10)/5);
    return 0;
}

int abs(int i)
{
    if(i<0)
        i = -i;
    return i;
}
```

```
#include <iostream>
using namespace std;

int abs(int i)
{
    if(i<0)
        i = -i;
    return i;
}

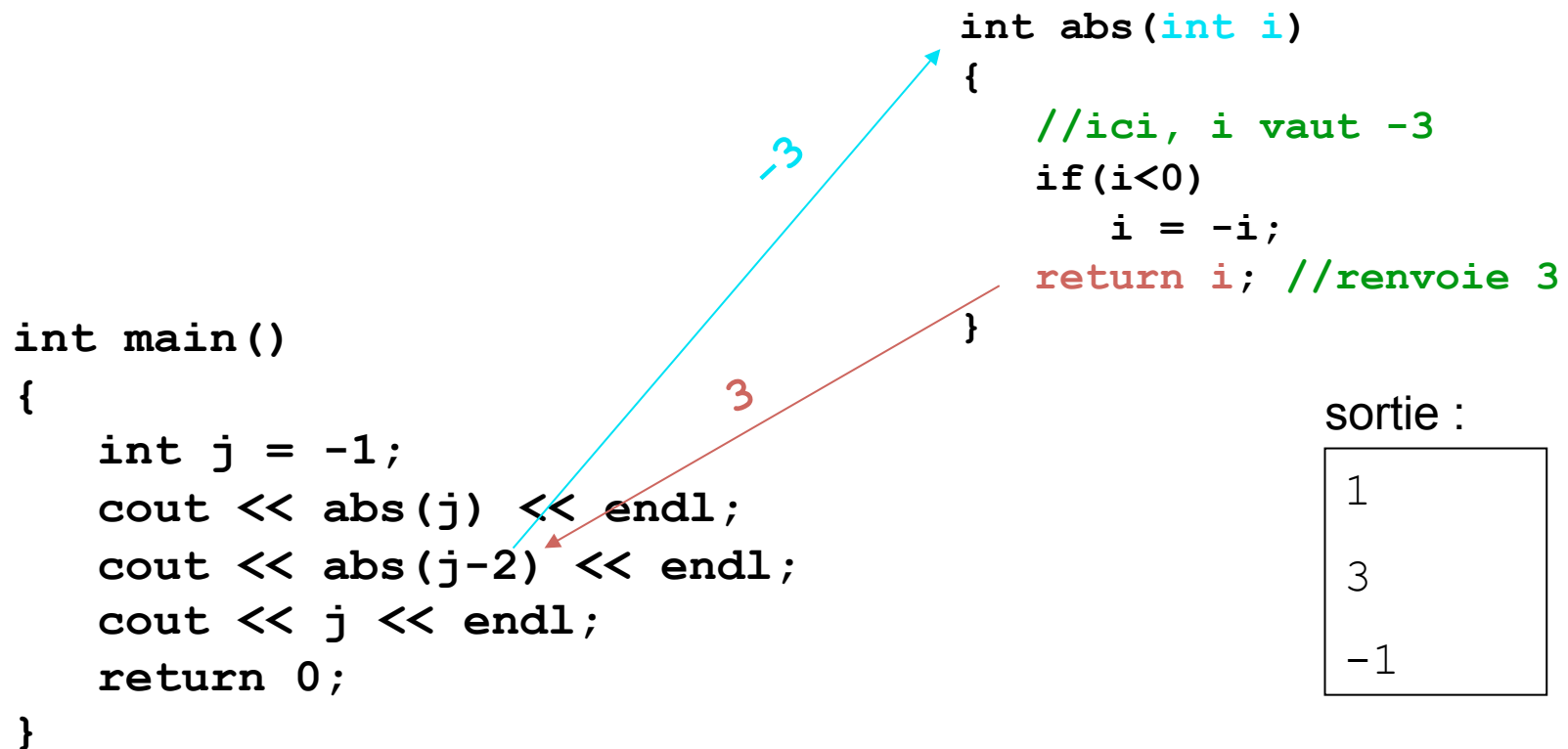
int main()
{
    int a;
    cin >> a;
    cout << abs(a) << endl;
    cout << abs(a-5) << endl;
    a = 2+(abs(a*10)/5);
    return 0;
}
```

Fonction `int main()`

- La fonction `int main()` est une fonction particulière appelée « *point d'entrée* » ou « *fonction principale* » du programme.
- Renvoie (avec `return`) un entier représentant le code d'erreur ou 0 par défaut.
- La fonction `main` a plusieurs prototypes :
 - `int main()`
 - `int main(int argc, char *argv[])`

Passage de paramètres

- **Par valeur** : l'expression donnée comme paramètre est évaluée et assignée à la variable locale.

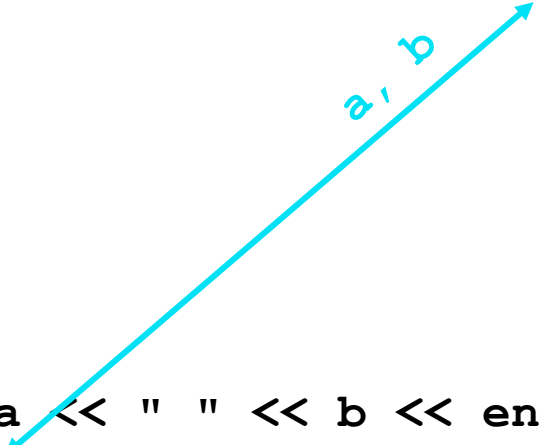


Passage de paramètres

- **Par référence** : la **variable** manipulée par la fonction appelée est celle donnée en paramètre.

```
int main()
{
    int a=1;
    int b=2;
    cout << a << " " << b << endl;
    swap(a,b);
    cout << a << " " << b << endl;
    return 0;
}

void swap(int& i, int& j)
{
    //ici, i est a et j est b
    int save = i;
    i = j;
    j = save;
}
```



sortie :

1	2
2	1

Portée

```
int abs(int i)
{
    if(i<0)
        i = -i;
    return i;
}
```

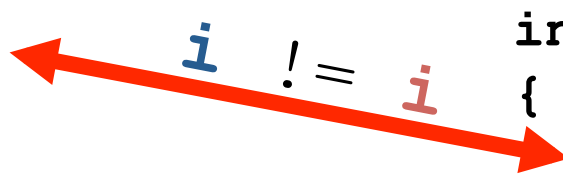
```
if(i<0)
    i = -i;
return i;
```

Portée de i

```
int main()
{
    int i = -1;
    i = abs(i);
    cout << i <<endl;
    return 0;
}
```

```
int i = -1;
i = abs(i);
cout << i <<endl;
return 0;
```

Portée de i



Structure d'un programme

1. include

```
#include <iostream>
```

2. using namespace

```
using namespace std;
```

3. Déclaration des constantes

```
const double EPS = 1.0e-8;
```

4. Déclaration des fonctions

```
int abs(int i);
```

5. Définition de la fonction main()

```
int main()
```

```
{
```

```
...
```

```
}
```

6. Définitions des fonctions

```
int abs(int i)
```

```
{
```

```
...
```

```
}
```

Remarques

- Ne pas définir une fonction dans une fonction.
- Ne pas définir de variables globales sauf constantes.
- Ne pas redéfinir les paramètres d'une fonction dans le corps de la fonction :

```
int fonction(int i)
{
    int i = 1;
    ...
}
```

- L'ordre des paramètres a de l'importance, pas leur nom.

Librairie `cmath`

- Librairie contenant des fonctions mathématiques.
- `#include <cmath>`
- Exemples :
 - `double cos(double x)`
 - `double sin(double c)`
 - `double sqrt(double x)`
 - `double pow(double base, double exponent)`
 - ...
- Remarque : n'utiliser cette librairie que si précisé dans l'énoncé.

Exercices

- En classe (séance 5) :
55, 54, 53, 56, 58, 60, 63
- Sur machine (séance 6) :
53, 59, 64, 65, 67, 68