Eléments de syntaxe du langage *C*, partie 2

Sébastien Jean

IUT de Valence
Département Informatique

v1.0, 9 octobre 2025

IUT Valence
Université Grenoble Alpes

Traduction en C des algorithmes en pseudo-code

Pseudo Code

```
FONCTION ma_fonction (n : entier) : entier

  ...

 RETOURNER ...

FIN FONCTION
```

IUT Valence
Université Grenoble Alpes

Traduction en C des algorithmes en pseudo-code

Programme C

```c
#include <stdio.h>

int ma_fonction(int n) {
 ...
 return ...
}

int main() {

 int n = 2;
 int resultat = ma_fonction(n);
 printf("resultat = %d\n",resultat);

 return 0;
}
```

- Par convention le programme est écrit dans un fichier `main.c`

Traduction en C du calcul du stock restant

- A partir de maintenant, chaque **fonction** sera écrite dans un répertoire différent (sous-répertoire de la racine du dépôt), portant le nom de la fonction (ou de sa variante) et contenant un sous-répertoire `src` où sera écrit le programme (dans le fichier `main.c`) et un sous-répertoire `build` où sera produit l'exécutable

IUT Valence
Université Grenoble Alpes

Traduction en C du calcul du stock restant

Pseudo Code

```
FONCTION stock_restant
(initial : entier, retrait : entier) : entier

 VARIABLE reste : entier

 reste ← intial - retrait
 RETOURNER reste

FIN FONCTION
```

- Pré-conditions : $initial \geq retrait \geq 0$
- Post-condition : Stock restant égal à $initial - retrait$

IUT Valence Université Grenoble Alpes

Traduction en C du calcul du stock restant

Code C

```c
#include <stdio.h>

int stock_restant(int initial, int retrait) {

 int reste = initial - retrait;
 return reste;

 // N.B. on peut se passer de la variable et écrire
 // return initial - retrait
}

 ... (suite après)
```

IUT Valence Université Grenoble Alpes

Traduction en C du calcul du stock restant

Code C

```
  ...

int main() {

  int stock = 100;
  int retrait = 10;
  printf("stock initial : %d\n", stock);
  printf("retrait : %d\n", retrait);
  printf("stock final : %d\n",
         stock_restant(stock, retrait));
}
```

IUT Valence Université Grenoble Alpes

Traduction en C du calcul du stock restant

- En considérant que les pré-conditions sont remplies (pas de traitement des cas d'erreur) :
  - **Définir** un **jeu d'essai** (dans un fichier `essai` à la racine du répertoire)
  - **Valider l'algorithme** avec le jeu d'essai

IUT Valence
Université Grenoble Alpes

Traduction en C du calcul de la surface d'un cercle

Pseudo Code

```
FONCTION surface (r : réel) : réel

 VARIABLE resultat : réel
 CONSTANTE PI : réel (3.14)

 resultat ← PI * r * r

 RETOURNER resultat

FIN FONCTION
```

- **Pré-condition** : $r \geq 0$
- **Post-condition** : surface égale à $\pi * r^2$

IUT Valence Université Grenoble Alpes

Traduction en C du calcul de la surface d'un cercle

Code C

```c
#include <stdio.h>

float surface(float r) {

 const float PI = 3.14159;
 float resultat = PI * r * r;
 return resultat;

 // N.B.
 // - on peut se passer de la variable et écrire
 //    return PI * r * r;
 // - on peut définir la constante globalement
 //    et/ou avec #define
}

 ... (suite après)
```

Traduction en C du calcul de la surface d'un cercle

Code C

```
  ...

int main() {

  float r = 2.0;
  printf("rayon : %f\n", r);
  printf("surface : %f\n",  surface(r));
  return 0;
}
```

Traduction en C du calcul de la surface d'un cercle

- En considérant que les pré-conditions sont remplies (pas de traitement des cas d'erreur) :
  - **Définir** un **jeu d'essai** (dans un fichier `essai` à la racine du répertoire)
  - **Valider l'algorithme** avec le jeu d'essai

IUT Valence
Université Grenoble Alpes

Traduction en C du calcul de la surface d'un cercle

- **Réécrire le programme** en utilisant la définition existante dans la librairie standard C (dans le fichier `math.h`) de la constante `M_PI`

IUT Valence Université Grenoble Alpes

Traduction en C du calcul de la surface d'un cercle

Code C

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

float surface(float r) {

 float resultat = M_PI * r * r;
 return resultat;

 // N.B. on peut se passer de la variable et écrire
 // return M_PI * r * r;
}

 ...
```

Fin !

IUT Valence
Université Grenoble Alpes