Algorithmique & Programmation en C

PROGRAMMER

Ressources :

Diapos PROGRAMMER

Objectif :

Définition de la programmation.

1. DEFINITIONS

Le programmeur crée des programmes par le processus de programmation.

a- Programme

Le dictionnaire Larousse précise : *« Un programme informatique est un ensemble d'instructions et de données représentant un algorithme et susceptible d'être exécuté par un ordinateur ».*

De façon générale, un **programme** définit des **traitements à appliquer à des informations**. Plus précisément, **un traitement est décrit par une suite d'instructions qui visent à manipuler, de façon ordonnée, des données numériques ou textuelles**. Une analogie peut-être faite entre un programme et une recette de cuisine. La recette décrit le mode opératoire pour obtenir un plat, c'est-à-dire la liste et l'ordre des opérations à effectuer pour transformer les ingrédients, dont nature et quantité ont été préalablement précisées. Les opérations correspondent aux instructions et les ingrédients aux données.

Les programmes sont écrits dans des langages informatiques compréhensibles par les ordinateurs. Nous nous intéressons aux programmes écrits en **langage C**, qui est un langage algorithmique et procédural, à exécution séquentielle :

- ✗ Un **langage algorithmique** permet la résolution d'un problème informatique par analyse descendante (décomposition d'un problème en sous-problème jusqu'à descendre à des actions primitives -simples instructions-), voir annexe de cette fiche.
- ✗ Un **langage procédural** organise ses traitements et données associées dans des sous-programmes appelés procédures ou fonctions.
- ✗ Une **exécution séquentielle** indique que l'ordinateur exécute les instructions élémentaires une par une dans l'ordre où elles se présentent et en suivant leur logique.

La famille des langages procéduraux/algorithmiques constitue la base de toute connaissance informatique en programmation. Cette approche permet d'appréhender les manipulations de base dans tous les programmes comme, par exemple, les programmes écrits en langage objet.

Les programmes peuvent prendre plusieurs formes, selon leur stade de développement et le formalisme dans lequel ils sont écrits :

- ✗ L'**Algorithme** est un programme en langage naturel (exemple : français) ;
- ✗ Le **Programme Source** est un programme en langage informatique (exemple : langage C) ;
- ✗ Le **Programme Exécutable** est un programme en langage binaire (directement interprétable par la machine).

b- Programmation

Le dictionnaire Larousse précise : *« La programmation est l'ensemble des activités liées à la définition, l'écriture, la mise au point et l'exécution de programmes informatiques »*.

Le processus de **programmation** permet au programmeur de **créer une solution informatique** exécutable par un ordinateur (programme ou application), **qui répond à un cahier des charges**.

Les étapes de développement itératives sont, essentiellement :

1. **Lecture attentive et Etude du cahier des charges**.
2. **Conception de la solution** : Analyse par approche descendante du problème de façon à obtenir les **algorithmes** des fonctions (instructions ordonnées et données). 45 %
3. **Traduction** des algorithmes **en langage C** en respectant les **Règles de Qualité** et **Documentation** du code source. 15 %
4. **Tests** du programme pour tous les points du cahier des charges et dans tous les cas prévisibles. 40 %

Remarque :

Entre les étapes de programmation 3 et 4, une étape est réalisée par l'ordinateur lorsque le programmeur choisit les opérations « Make » ou « Run » de l'environnement de développement intégré (EDI Code::Blocks ou Visual C++ Express, ...). Il s'agit de la traduction du code source en code binaire (étapes de **compilation** et d'**édition de liens** -link-). Quelques éléments matériels l'exécution des programmes doivent être connus :

- ✗ La **RAM** (Random Access Memory) : mémoire de travail de l'ordinateur, qui sert à charger temporairement toutes les applications en cours d'exécution, dont le programme que vous êtes en train d'exécuter (instructions et données codées en binaire). Cette mémoire est volatile : lorsque l'ordinateur n'est pas alimenté, elle est vidée.
- ✗ Le **CPU** (Central Processing Unit) : unité de traitement des programmes chargés dans la RAM. Il exécute les instructions une par une, selon l'ordre et la logique d'exécution, et modifie les données impliquées dans ces instructions. Le CPU contient l'**ALU** (Arithmetic and Logic Unit) qui est l'unité de calcul de l'ordinateur.

2. ALGORITHMES

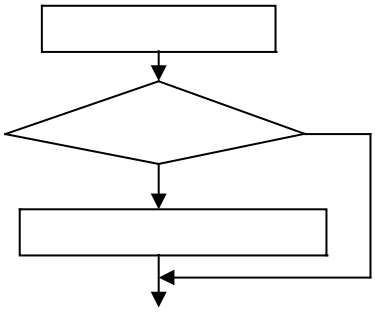
Un **algorithme** précise, en LANGAGE NATUREL, les données et les instructions nécessaires, ainsi que l'enchaînement de ces dernières, afin de réaliser une tâche donnée.

a- Présentation des Algorithmes

Nous adopterons le formalisme suivant pour nos algorithmes :

```
/*****************************************************************************
        Commentaire d'en-tête d'algorithme
*****************************************************************************/

//
// Commentaire d'en-tête de fonction
//
ALGO  Nom_Fonction1
            VAR          nomVariables :  types            // Commentaire de données
             CONST       NOMCONSTANTES :  types     // Commentaire de données
DEBUT
             // Commentaire de partie
             Instruction1
             Instruction2
             // Commentaire de partie
             Instruction3
             Instruction4
FIN

//
// Commentaire d'en-tête de fonction
//
ALGO  Nom_Fonction2
            VAR    nomVariables :  types                // Commentaire de données
DEBUT
             // Commentaire de partie
             Instruction5
FIN
```

Remarque : des parties d'algorithmes peuvent aussi être représentés avec des graphiques, de type organigrammes :



b- Intérêt des Algorithmes

Les phases d'Analyse du cahier des charges et de Conception occupent une grande partie du temps de développement d'une application (autour de 45%) ; ce sont donc des phases importantes pour la construction des programmes : il ne suffit pas de se précipiter pour aligner des lignes de code. La conception comprend la création d'algorithmes prévisionnels du programme.

La réalisation préalable d'algorithmes permet de :

- ✗ se concentrer sur la recherche d'une solution logique ; en effet, en utilisant un langage naturel et simple, le programmeur s'affranchit des aspects syntaxe d'un langage informatique et ne bute donc pas sur la syntaxe pointilleuse du langage de programmation.
- ✗ planifier le développement par étape ; en effet, la réflexion préalable sur papier permet d'avoir une vue d'ensemble de la solution et d'éviter d'avoir à recommencer un programme trop vite et mal parti !
- ✗ documenter le travail ; en effet, la traduction des algorithmes permet de commenter rapidement son code et de produire facilement, par la suite, des rapports de conception.

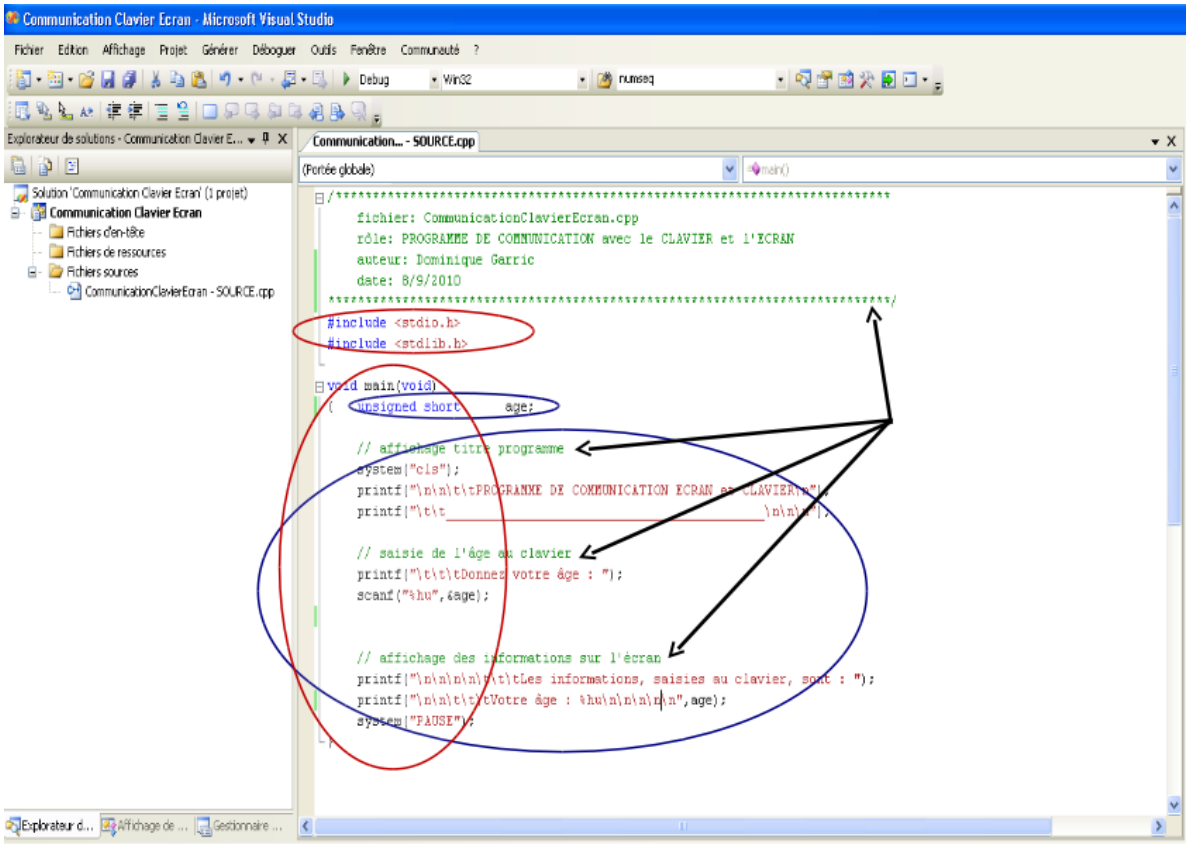
En conséquence, les **gains sur le processus de développement résultants du temps passé à établir des algorithmes** sont listés ci-dessous :

- ✗ Gain en temps de développement global ;
- ✗ Gain en qualité de développement ;
- ✗ Gain en planification des tests ;
- ✗ Gain en efficacité de maintenance et de mise à jour.

3. CODAGE EN LANGAGE C

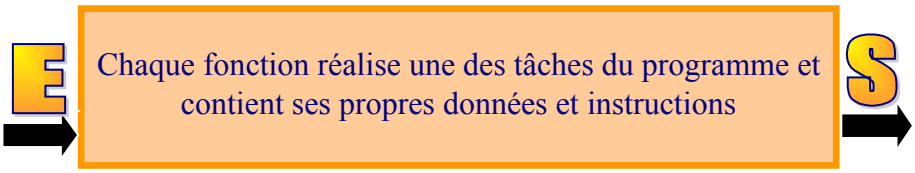
La structure générale d'un fichier source en C est constituée d'inclusion de bibliothèques (contenant les fonctions standards directement utilisables), puis de fonctions successives (dont la principale est main()). A l'intérieur de la fonction main(), se trouvent d'abord la partie donnée, puis la partie traitement. Divers types de commentaires doivent être rajoutés dans le code :

```
/*************************************************************************
            Commentaire d'en-tête de fichier source :
Rôle fichier : …                    Nom du fichier : ...
Auteur : …                          Date : …
*************************************************************************/
#include <stdio.h>

//
// Commentaire d'en-tête de fonction
//
int main()
{       Instructions de déclaration de données ;      // Commentaire de données

        // Commentaire de partie
        Instruction1 ;
        Instruction2 ;

        // Commentaire de partie
        Instruction3 ;
        Instruction4 ;
}
```



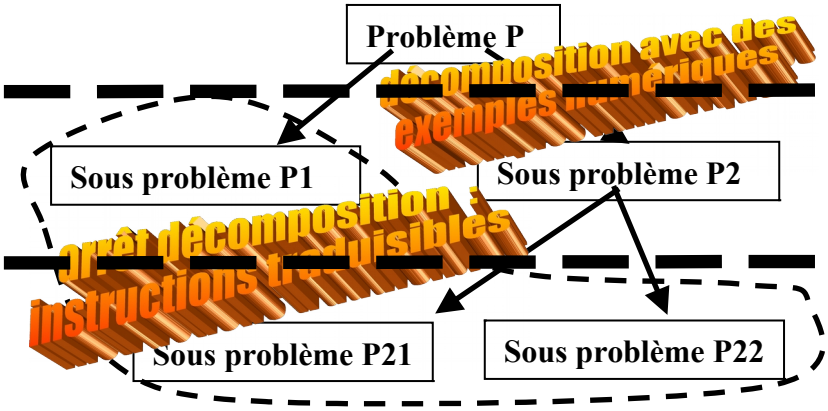
ANNEXE : ANALYSE PAR APPROCHE DESCENDANTE

L'analyse du cahier des charges par approche descendante permet d'obtenir les algorithmes des fonctions du programme:



La méthode pour approcher la solution progressivement consiste à partir d'un problème initial (cahier des charges) et à arriver à une solution finale (ensemble d'algorithmes directement traduisibles) en passant par des niveaux intermédiaires. Il s'agit de recommencer, tant que les sous-problèmes ne contiennent pas des instructions directement traduisibles dans le langage de programmation, de :

- ✗ décomposer les sous problèmes en nouveaux sous problèmes moins complexes.
- ✗ utiliser des exemples numériques pour aider à décomposer, pas à pas, le raisonnement.



ALGO problème P
(niveau 1)
DEBUT
sous problème P1
sous problème P2
FIN

ALGO problème P1
(niveau 2)
DEBUT
instruction1
instruction2
FIN

ALGO problème P2
(niveau 2)
DEBUT
sous problème P21
sous problème P22
FIN

ALGO problème P21
(niveau 3)
DEBUT
instructions
FIN

ALGO problème P22
(niveau 3)
DEBUT
instructions
FIN