Langage procédural

Le langage C

Nizar OUARTI

Laboratoire ISIR (email: ouarti@isir.upmc.fr)

2012

Qu'est ce qu'un algorithme

- Ici nous n'étudierons que les algorithmes séquentiels
- Chaque instruction se fait l'une à la suite de l'autre
- L'algorithme sert à résoudre un problème grâce à un jeu d'instructions fini et de façon séquentielle
- Un des premiers exemples d'algorithmes : Résolution des équations du second degré ;
- Al Khawarizmi au IXième siècle, la latinisation de son nom a donné algorithme.

Ecrire un algorithme

- Extraction des racines des équations du second degré
- Les solutions de $ax^2 + bx + c = 0$
- utiliser :
  - if (Si : condition)
  - $\Delta = b^2 - 4ac$
- Essayez de penser à tous les cas

Plusieurs cas de figures

- Les racines représentent l'endroit où la courbe qui représente l'équation passe par l'axe des abcisses
- Voir la figure 1
  - if $\Delta == 0$ Il n'y a qu'une seule racine $\{x_1\} = \frac{-b}{2a}$
  - if $\Delta > 0$ Il y a deux racines $\{x_1, x_2\} = \frac{-b \pm \sqrt{\Delta}}{2a}$
  - if $\Delta < 0$ Il n'y a pas de racines (réelle) $\{x_1^{im}, x_2^{im}\} = \frac{-b \pm i\sqrt{|\Delta|}}{2a}$ (avec $i = \sqrt{-1}$)

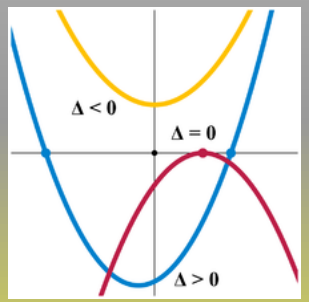

FIGURE: Différentes courbes en fonction de la valeur de Δ

algorithme : x1,x2=khawarizmi(a,b,c)

```
1  if delta==0
2  {
3          x1= -b/2a;
4          str='Nul';
5  }
6  if delta>0
7  {
8          x1= (-b + sqrt(delta))/2a;//fonction sqrt
9          x2= (-b - sqrt(delta))/2a;
10         str='positif';
11 }
12 if delta<0
13 {
14         x1= (-b + i sqrt(abs(delta)))/2a;//fonction abs
15         x2= (-b - i sqrt(abs(delta)))/2a;
16         str='négatif';
17 }
18 Ecrire('Delta est:');Ecrire(str);
19 Ecrire('Mon premier algorithme est fini');
```

Amélioration de l'algorithme

- Que manque-t-il pour que l'algorithme soit vraiment fonctionnel ?

Amélioration de l'algorithme

- Que manque-t-il pour que l'algorithme soit vraiment fonctionnel ?
- Il manque une condition testant si a=0
- Dans le cas a=0, nous aurions eu une division par zéros
- if a==0 $\longmapsto$ $bx = -c \longmapsto x_1 = -c/b$
- else $\longmapsto$ exécuter {x1,x2}=khawarizmi(a,b,c)

Amélioration de l'algorithme

- Que manque-t-il pour que l'algorithme soit vraiment fonctionnel ?
- Il manque une condition testant si a=0
- Dans le cas a=0, nous aurions eu une division par zéros
- if a==0 $\longmapsto$ $bx = -c \longmapsto x_1 = -c/b$
- else $\longmapsto$ exécuter {x1,x2}=khawarizmi(a,b,c)
- Est ce que cette fois tous les cas de figure ont été traités ?

Amélioration de l'algorithme

- Que manque-t-il pour que l'algorithme soit vraiment fonctionnel ?
- Il manque une condition testant si a=0
- Dans le cas a=0, nous aurions eu une division par zéros
- if a==0 $\longmapsto$ $bx = -c \longmapsto x_1 = -c/b$
- else $\longmapsto$ exécuter {x1,x2}=khawarizmi(a,b,c)
- Est ce que cette fois tous les cas de figure ont été traités ?
- Non : Dans l'embranchement if a==0, il fallait traiter le cas if b==0 $\longmapsto$ Ecrire (Pas de solution)

Comment écrire un algorithme

- Commentaires sur le code
- Déclaration des variables $\longmapsto$ typage
- Initialisation des variables $\longmapsto$ leur donner une valeur initiale
- Instructions : les unes après les autres
- Découpage en sous-parties fonctionnelles : procédures et fonctions
- Minimiser le temps de calcul et la place en mémoire
- Ecrire le résultats des calculs dans un fichier ou en extraire des données
- Et bien sûr la réflexion...

Commentaires : à quoi cela sert-il ?

- Les commentaires n'ont aucune fonction et paradoxalement sont **très importants**
- Les commentaires servent à donner des informations sur le fonctionnement d'un programme
- Ils peuvent expliquer le rôle d'une variable ou d'une partie du code
- Les commentaires prennent leur importance lorsque
  - Le code est long, on ne se souvient plus de tout
  - On reprend son propre code longtemps après
  - On reprend le code d'un autre

Algorithme et commentaires

Si $A == B$ // Ici les commentaires du code.

Différents commentaires

- Il existe deux types de commentaires en C
- Les commentaires 'unilignes' : toutes les instructions à droite ce signe sont commentées sur la même ligne
- Les commentaires 'multilignes' : toutes les instructions entres les deux signes sont commentées

Commentaires simples

```
1 if (A==B) //Commentaires
2 {
3         A=2*A;
4 }
```

Commentaires multiples

```
1 if (A==B)
2 /* Commentaires
3 {
4         A=2*A;
5 }
6
7 */
```

Commentaires et débugage

- Lorsqu'une ligne est commentée, elle n'est pas prise en compte pour la compilation
- Bien qu'ils n'aient pas été créés expressément pour ça les commentaires sont un parfait outil pour le débugage
- Ils permettent de commenter la ligne qui semble poser problème dans le programme
- Ils permettent d'enlever des parties de code mais sans les effacer définitivement
- Une fois l'erreur détectée le code sera restauré en décommentant simplement
- Autre possibilité compilation conditionnelle

Données dans un ordinateur

- Les données sont stockées sous forme de 0 et 1 : des bits.
- Ce stockage est dit sous forme binaire
- Comment coder un nombre ou un caractère sous cette forme ?
- Les variables peuvent être vues comme des containers de stockage

Pour les nombres

- Combien de bits faut-il pour coder des nombres compris entre 0 et 255
- En binaire pour n bits disponibles on a 2^n combinaisons
- Dans notre cas avec n=8 on a 256 combinaisons
- Avec 8 bits (1 octet) on peut coder les nombres compris entre 0 et 255
- Comment faire pour coder un nombre allant de 0 à ∞ ?
- Comment faire si ce nombre n'est pas un entier mais un réel ?

Pour les caractères

- Un jeu de 256 caractères appelé ASCII (American Standard Code for Information Interchange) a été défini comme standard.
- Chaque caractère représente un nombre binaire
- De combien de bits a-t-on besoin pour coder un caractère ?

Les types de Données

- Comme nous l'avons vu un nombre ne peut pas se stocker en mémoire de la même manière qu'un caractère
- Il serait aussi intéressant au sein des nombres de différencier les nombres à virgule des nombres entiers
- L'intérêt de l'usage de container de stockage est l'économie. Pas besoin d'un immense garage pour ranger uniquement son vélo...
- Pourquoi utiliser la place en mémoire plus importante d'un nombre à virgule si on a besoin d'utiliser que des valeurs entières
- Imaginons un cas extrême aussi où une variable ne pourrait être que vrai ou fausse (0 ou 1)
- Les **types de données** ont été inventés pour cela

Les types de Données

- Les types de données aident à structurer la pensée et le code en sachant ce que l'on manipule : un entier, un booléen, un nombre à virgule ou un caractère.
- De plus, ils permettent le stockage à l'économie des variables.
- En effet, rien de plus simple que de stocker un 0 ou 1 dans la place requise par un nombre entier mais ce n'est pas très économe en place mémoire...

Tableau des types de Données

Type de donnée	Explications	octets	intervalle de valeurs
signed char	Caractère	1	-128 à 127
char	Caractère	1	0 à 255
short int	Entier court	2	-32768 à 32767
unsigned short int	Entier court non signé	2	0 à 65535
int	Entier	4	-2147483648 à 2147483647
unsigned int	Entier non signé	4	0 à 4294967295
long int	Entier long	4	-2147483648 à 2147483647
unsigned long int	Entier long non signé	4	0 à 4294967295
float	Réel	4	$3.4\ 10^{-38}$ à $3.4\ 10^{38}$
double	Réel double précision	8	$1.7\ 10^{-308}$ à $1.7\ 10^{308}$
long double	Réel double long	10	$3.4\ 10^{-4932}$ à $3.4\ 10^{4932}$

TABLE: Tableau récapitulatif des types de données. Certains éléments peuvent changer en fonction du type de processeur. Ex : processeur 16 ou 32 bits.

Initialisation de variables

- Il suffit d'affecter une valeur à la variable
- L'affectation se fait avec un objet de même type

```
1  int x;              // Déclaration de la variable
2  x=4                 // Affection de la variable
3
4  char letter;        // Déclaration de la variable
5  letter='b' ;        // Affectation type identique
6
7  bool flag=true;     // Déclaration et affectation
8
9  char letter2;       // Déclaration de la variable
10 letter=letter2;     // Affectation de la valeur
11                     // dans une autre de même type
```

Charte à suivre

- Granularité
  - Une fonction ne doit pas dépasser 25 lignes
  - Chaque ligne ne doit pas dépasser 80 caractères
  - Une fonction donnée ne doit résoudre qu'un seul problème (unité fonctionnelle)
- Fonction commenté au format doxygen
- Le code doit être indenté avec accolade fermante et ouvrante au même niveau
- Les noms de variables et fonctions explicites, voire mémotechniques : pas 'toto', 'n' ou 'x'
- Les constantes seront définies avec des #define et seront écrites en majuscule
- les noms de variables composés devront être séparé par des underscore :'_'
- les variables static si elles existent doivent être précédée de st_

Problèmes avec les données

- Problème : On voudrait pouvoir gérer le dossier d'un patient
- Différentes variables composent son dossier
- Son nom, son prénom, son groupe sanguin, s'il est marié ou non, son sexe, son âge, le nombre d'enfants qu'il a
- On voudrait aussi gérer son taux de globules rouges et blancs dans le sang, ainsi que le nombre de plaquettes
- Comment représenteriez-vous ces données dans l'ordinateur ?

Problèmes avec les données

- Problème : On voudrait pouvoir gérer le dossier d'un patient
- Différentes variables composent son dossier
- Son nom, son prénom, son groupe sanguin, s'il est marié ou non, son sexe, son âge, le nombre d'enfants qu'il a
- On voudrait aussi gérer son taux de globules rouges et blancs dans le sang, ainsi que le nombre de plaquettes
- Comment représenteriez-vous ces données dans l'ordinateur ?
- Ecrivez un algorithme qui détecte si un patient est une femme et si elle a un taux de globules rouges en dessous d'un seuil donnée et si son âge est supérieur à 65 ans alors on lance une alerte en précisant son identité et en pensant à joindre le conjoint.

Problèmes avec les données

- Problème : On voudrait pouvoir gérer le dossier d'un patient
- Différentes variables composent son dossier
- Son nom, son prénom, son groupe sanguin, s'il est marié ou non, son sexe, son âge, le nombre d'enfants qu'il a
- On voudrait aussi gérer son taux de globules rouges et blancs dans le sang, ainsi que le nombre de plaquettes
- Comment représenteriez-vous ces données dans l'ordinateur ?
- Ecrivez un algorithme qui détecte si un patient est une femme et si elle a un taux de globules rouges en dessous d'un seuil donnée et si son âge est supérieur à 65 ans alors on lance une alerte en précisant son identité et en pensant à joindre le conjoint.
- Que feriez-vous si vous avez 200 patients à gérer ?