

Algorithmique et structures de données 2017-18

Contenu du cours 1 du 21.09.2017

1. Présentation personnelle : Paul Albuquerque, resp. de filière
bureau B410
022 546 2554
paul.albuquerque@hesge.ch
2. Organisation du module (2 cours, 50 % chacun)
 - Algorithmique et structures de données
 - 1^{er} semestre
 - bases de la programmation en Ada jusqu'à Noël
 - algorithmique jusque fin janvier
 - 2^{ème} semestre
 - algorithmique
 - Au moins 2 évaluations par semestre
 - Programmation séquentielle
 - 1^{er} semestre : travaux pratiques en langage Ada
 - 2^{ème} semestre : travaux pratiques en langage C

3. Qu'est-ce qu'un algorithme ?

Informellement, c'est une recette ou une marche à suivre

Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.

Le mot *algorithme* vient du nom latinisé du mathématicien perse Al-Khawarizmi

Les algorithmes des grecs : Euclide, Erastosthène

Le père de l'algorithmique : **Muhammad Ibn Mūsā al-Khwarizmi**

Résolution d'un problème

- décomposition en sous-problèmes
- résolution des sous-problèmes
- assemblage des résultats

4. Notions de base d'algorithmique séquentielle
 - Définition d'une variable
 - Séquences d'instructions
 - Boucles et tests
5. Algorithme de vérification qu'un nombre est 1^{er}
 - Comment demander le nombre
 - Forme de l'algorithme (boucle et tests)
 - Condition d'arrêt de la boucle
6. Qu'est qu'un programme ADA ?
 - Clause de contexte
 - Entête du programme
 - Partie déclarative
 - Bloc d'instructions

7. Les types de base : les entiers (Integer)

- numérotation binaire en particulier sur 32 bits
- nombres négatifs => complément à 2
- numération sur 3 bits :

000	0	100	-3
001	1	101	-2
010	2	110	-1
011	3	111	-4

Opérations particulières :

- Exponentiation : $A ** B$ A est entier, B doit être entier positif
- Division entière : $5 / 2 = 2$, $10 / 3 = 3$, $18 / 5 = 3$
- Reste de la division entière : $5 \text{ rem } 2 = 1$, $10 \text{ rem } 3 = 1$, $18 \text{ rem } 5 = 3$

Attributs du type Integer : Integer'First / Integer'Last
Integer'Succ / Integer'Pred

8. Les types de base : les réels (Float)

Float : écriture, opérations, exemples complets d'exponentiation

$A ** B$, en Ada le nombre A est un float et B est un entier => le résultat est un float !

Sur 32 bits : 1 bit de signe, 8 bits pour l'exposant et 23 bits pour la mantisse.

Exemple : coder 19,625 en binaire

$19 : 10011 = 2^4 + 2^1 + 2^0$; $0,625 : 0,101 = 2^{-1} + 2^{-3}$

$19,625 : 10011,101 = 0,10011101 * 2^5$

Le signe du nombre est stocké sur dans le 1^{er} bit (0 positif / 1 négatif).

L'exposant est stocké sur 8 positions => 256 valeurs => -128..127. Donc 5 = 00000101

La mantisse de 23 bits stockée est 001110100000000000000000

(le 1^{er} 1 est omis car obligatoire)

19,625 stocké en binaire : 0 00000101 001110100000000000000000

9. Structures de boucle :

loop

Instructions; (parmi elles éventuellement exit when condition;)

end loop;

for I in *intervalle* loop

Instructions;

end loop;

while *condition* loop

Instructions

end loop;

10. Structures de test :

If *condition* then

Instructions;

elsif *condition* then

Instructions;

else

Instructions;

end if;

11. Programme complet : PGCD de deux nombres