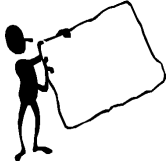


Mémo n. 1  
LE LANGAGE DU I32  
I32 Preuves et Analyses d'algorithmes  
P. Véron

1. LES TYPES SCALAIRES PRÉDÉFINIS



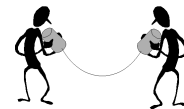
- entier
- réel
- caractère
- chaîne de caractères
- booléen
- tableau.

2. LES OPÉRATEURS

- +, -, ×, div, mod, [], []. L'opérateur + est valable pour les chaînes de caractère : 'aa'+ 'bb' = 'aabb'.
- opérateurs de comparaison : <, >, ≤, ≥, ≠.
- opérateurs logiques : non, et, ou.

3. GESTION DES ENTRÉES-SORTIES

- lire(a), a variable de type scalaire
- écrire(a), a variable de type scalaire



4. STRUCTURE GÉNÉRALE D'UN ALGORITHME

```
1 Algo ???
2 données
3   identificateur1 : type1
4    $\vdots$ 
5   identificateurN : typeN
6   < déclaration des fonctions >
7   début
8     < instruction(s) >
9   fin
```

5. STRUCTURE D'UNE FONCTION

fonction *nom\_de\_la\_fonction*(*id1* (ou *id1*) : *type1*, ..., *idN* (ou *idN*) : *typeN*)

```
données
  identificateur1 : type1
   $\vdots$ 
  identificateurN : typeN
début
  < instruction(s) >
  retourner(valeur)
fin
```

*valeur* étant de type entier, réel, booléen, caractère ou chaîne de caractères.

## 6. AFFECTATION

L'affectation se fait en utilisant le symbole  $\leftarrow$ .

Exemple :  $A \leftarrow B$ ,  $A \leftarrow 3$ ,  $A \leftarrow A + 1$ .

## 7. CONSTANTES PRÉDÉFINIES

Le langage comporte deux constantes booléennes prédéfinies : VRAI et FAUX.

## 8. STRUCTURES DE CONTROLE

- Structure itérative

**tant que** *condition booléenne* **faire**  
                   < *instruction(s)* >  
**fintq**



- Choix simple

**si** *condition booléenne* **alors**  
   < *instruction(s)* >  
**finsi**

- Choix avec alternative

**si** *condition booléenne* **alors**  
   < *instruction(s)1* >  
   **sinon**  
   < *instruction(s)2* >  
**finsi**



- Choix multiple

**si** *condition booléenne1* **alors**  
   < *instruction(s)1* >  
**sinon si** *condition booléenne2* **alors**  
   < *instruction(s)2* >  
 ⋮  
**sinon si** *condition booléenneN* **alors**  
   < *instruction(s)N* >  
**finsi**

OU

**si** *condition booléenne1* **alors**  
   < *instruction(s)1* >  
**sinon si** *condition booléenne2* **alors**  
   < *instruction(s)2* >  
 ⋮  
**sinon**  
   < *instruction(s)N* >  
**finsi**

**selon que identificateur soit**

*valeur1* : *instruction(s)1* (ou rien)

*valeur2* : *instruction(s)2* (ou rien)

⋮

*valeurN* : *instruction(s)N* (ou rien)

**finselon**