

*I*nformatique

4^{ème} année Scientifique

Professeur : Mohamed TRABELSI

Plan du cours

Algorithmique et Programmation

Chapitre n°1 : Les structures de données (6h)

- Leçon 1 : Les variables et les types standard de données
- Leçon 2 : Les expressions
- Leçon 3 : Le type scalaire énuméré et le type intervalle
- Leçon 4 : Les tableaux à une dimension

Chapitre n°2 : Les actions élémentaires simples (2h)

- Leçon 1 : L'affectation
- Leçon 2 : Les opérations d'entrée / sortie

Chapitre n°3 : Les structures de contrôle conditionnelles (4h)

- Leçon 1 : La structure conditionnelle simple
- Leçon 2 : La structure conditionnelle généralisée
- Leçon 3 : La structure conditionnelle à choix

Chapitre n°4 : Les structures de contrôle itératives (12h)

Chapitre n°5 : Les sous-programmes (16h)

Chapitre n°6 : Les traitements avancés (10h)

Exercice de révision

Enoncé :

On se propose de permuter 2 variables de type entier. Présenter une analyse puis un algorithme pour résoudre ce problème.

Méthode 1 : En utilisant une 3^{ème} variable.

Méthode 2 : Sans une 3^{ème} variable.

Traduction en Pascal (voir fichier : permut.pas)

Chapitre n°1 :

Les structures de données

Objectifs du cours :

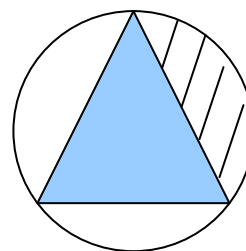
- Apprendre à manipuler les données dans un programme.
- Structures utilisées : Constante, variable et tableau (vecteur).
- Connaître la notion de type de données.

Leçon 1

Structures et types de données

I – Les constantes et les variables

Activité : Ecrire un programme qui calcul et affiche la surface de la partie hachurée du cercle suivant :



Le rayon du cercle et le côté du triangle équilatéral sont des données.

Pré analyse :

Surface hachurée : $\frac{s_{\text{cercle}} - s_{\text{triangle}}}{3}$

Surface du cercle : πr^2

Surface du triangle équilatéral : $\frac{\sqrt{3}}{4} c^2$

Rappel : $\sin \alpha = \frac{\text{opp}}{\text{hyp}}$
 $\sin 60 = \frac{\sqrt{3}}{2}$

Analyse :

Nom : Surface		
S	L. D. E	O.U
6	Résultat = écrire ("L'aire de la surface hachurée est : ", s)	s_hachrée
5	s_hachurée ← s_cercle – s_triangle / 3	s_cercle, s_triangle
3	s_triangle ← SQRT (3) / 4 * SQR (c)	c
4	s_cercle ← pi * SQR (r)	pi, r
1	r = Donnée (" Saisir le rayon du cercle : ")	
2	c = Donnée (" Saisir le côté du triangle : ")	
	pi = 3,14 (constante)	
7	Fin Surface	

T.D.O

Objets	Type	Rôle
s_hachrée	réel	
s_cercle,	réel	
s_triangle	réel	
c	réel	
r	réel	
pi	Constante = 3,14	

Traduction en Pascal (voir fichier : surface.pas)

Une constante est caractérisée par un nom et une valeur qui ne change pas tout au long de l'exécution du programme.

Déclaration :

En analyse :

T.D.O

Objets	Type	Rôle
Nom_constant	Constante = valeur	

En Pascal :

CONST Nom_constant = valeur ;

- Une variable est un objet dont la valeur peut être modifiée tout au long de l'exécution du programme. Elle est caractérisée par :

- Son nom
- Son contenu (la valeur)
- Son Type
- Sa taille en mémoire (en octet)

Déclaration :

En analyse :

T.D.O

Objets	Type	Rôle
Nom_variable	Type_variable	

En Pascal :

VAR Nom_variable : Type_variable ;

(Voir activité séquence d'affectation livre page 9)

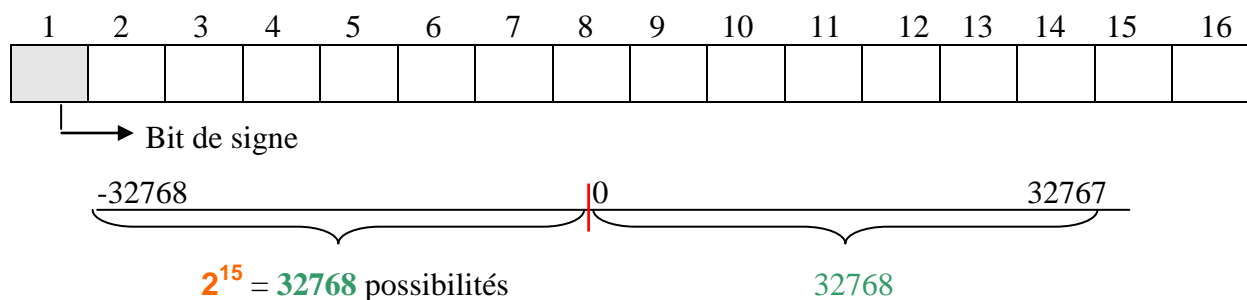
II – Les types de données

(Lecture page 10)

1. Les types numériques

a. Les entiers :

- Le type entier permet de manipuler des valeurs de l'ensemble \mathbb{Z}
- Type entier (*Integer*)
- Taille en Mémoire centrale : 2 octets.
- Domaine des valeurs : entre -32768 et 32767.
- Autres types entiers prédéfinis (livre page 11)



b. Les réels :

- Le type réel permet de manipuler des valeurs de l'ensemble \mathbb{R}
- Type réel (*Real*)
- Taille en Mémoire centrale : 6 octets.
- Domaine des valeurs : -10^{38} à 10^{38} .
- Format scientifique $2.0000000000E+3 \Leftrightarrow 2.10^3 \Leftrightarrow 2000.$
- Autres types réels prédéfinis (Help Turbo Pascal)

c. Les fonctions arithmétiques standards :

Voir livre page 13.

2. Le type booléen (Boolean)

- Domaine des valeurs : Deux valeurs logiques *vari* ou *faux*. (*True* or *false*)

3. Le type caractère (char)

(Lecture page 16)

- Une variable de type caractère occupe un octet en mémoire.
- A chaque caractère correspond un code ASCII qui est un entier entre 0 et 255.

(Voir table ASCII page 229)

- Notation : En analyse → "A", En Pascal → 'A'
- Opérateurs relationnels : On peut comparer les caractères entre eux.
Exemples : "a" ≠ "b" → Vrai, "A" > "a" → Faux
- Les fonctions prédéfinies : (Voir livre page 17).

Exemples :

Ord("A") = 65,	Pred("b") = "a"
Chr(122) = "z",	Majus("c") = "C",
Succ("a") = "b",	Majus("D") = "D" ,

4. Le type chaîne de caractères (string)

- Une variable de type chaîne de caractère est une suite ordonnée de caractère.
- Capacité : 0 à 255.
- Notation : En analyse → "L'algorithmique", En Pascal → ' L'algorithmique '
- Lors de la déclaration d'une chaîne on peut préciser la taille voulue.
- Exemple : En analyse

Objet	Type
Nom	Chaîne [20]

En Pascal

var nom : string[20];

- Une chaîne peut être manipulée caractère par caractère grâce à la notation nom[i]. i est l'indice (rang) du caractère voulu.

Exemple : nom ← "Ben Foulen" ⇔ nom[5] contient "F"

nom[4] ← "_" ⇔ nom = "Ben_Foulen"

- Fonctions et procédures prédéfinies sur les chaînes: Voir livre page 19.

Exemples :

ch1 ← "séden", ch2 ← "taire"

ch3 ← **concat** (ch1, ch2) ch3 contient "sédentaire"

L ← **long** (ch3) L contient 10

ch4 ← **sous-chaîne** (ch3, 3 ,8) ch4 contient "dentaire"

P ← **pos** ("aire", ch4) P contient 5

P ← **pos** ("BAC", ch4) P contient 0

Efface (ch3, 1, 6)

ch3 contient "aire"

Insert (" ", ch4, 5)

ch4 contient "dent aire"

Soit la variable *date* de type réel et $date \leftarrow 1,618$ **ConvCh** (date, ch5)

ch5 contient "1,618"

Soient a et erreur 2 variables du type entier et $ch6 \leftarrow "123"$ **valeur** (ch6, a, erreur) \Leftrightarrow a contient 123 erreur = 0**valeur** (ch4, a, erreur) \Leftrightarrow a contient 0 erreur = 1

- Correction exercice n°12 page 23.

a. Analyse

Nom : min_maj		
S	L. D. E	O.U
3	Résultat = écrire ("Après changement de la casse : ", maj)	maj
2	maj = [maj ← ""] Pour i de 1 à long (ch) faire Si ch[i] dans ["a".."z"] alors maj ← maj + Chr (Ord (ch[i]) – 32) sinon maj ← maj + ch[i] Fin si Fin pour	i, ch
1	ch = donnée ("Donner une chaîne : ")	
	i = compteur	
4	Fin min_maj	

T.D.O

Objets	Type	Rôle
ch	Chaîne	
maj	Chaîne	
i	Octet	Compteur

b. Algorithme

- 0) Début min_maj
- 1) Ecrire ("Donner une chaîne : "), (Lire (ch))
- 2) maj ← ""
Pour i de 1 à long (ch) faire
 Si ch[i] dans ["a".."z"] alors
 maj ← maj + Chr (Ord (ch[i]) – 32)
 sinon
 maj ← maj + ch[i]
 Fin si
Fin pour
- 3) écrire ("Après changement de la casse : ", maj)
- 4) Fin min_maj

c. Pascal (voir fichier : min_maj.pas)

Leçon 2

Les expressions

I – Les opérandes

Une expression simple est constituée d'opérandes reliés par des opérateurs. On distingue deux types :

- Les expressions arithmétiques exemple : $54 \text{ MOD } 5 + \text{carré}(4)$
- Les expressions logiques comme : $("a" > "A")$

II – Les opérateurs

1. Les opérateurs arithmétiques :

Opérateur	Type
+	Réel, entier
-	Réel, entier
*	Réel, entier
DIV	Entier
MOD	Entier
/	Réel, Entier (seulement sur les opérandes, le résultat est réel)

2. Les opérateurs logiques :

Opérateur	Type
Non (Not)	Booléen
Et (And)	Booléen
Ou (Or)	Booléen
OuEx (xor)	Booléen

Table de vérité

A	B	Non (A)	A ET B	A OU B	A OUex B
F	F	V	F	F	F
F	V	V	F	V	V
V	F	F	F	V	V
V	V	F	V	V	F

3. Les opérateurs relationnels :

Analyse	<	≤	=	≥	>	≠	Dans
Pascal	<	<=	=	>=	>	<>	IN

Les opérateurs relationnels sont utilisables avec tous les types prédéfinis.

III – Évaluation d'une expression

- Ordre de priorité des opérateurs arithmétiques : (), −, / **même priorité que** ×, +
- Ordre de priorité des opérateurs logiques : Non, Et, Ou **même priorité que** OuEx

Activité : évaluer les expressions suivantes :

- ("Grand" < "petit") ET ("PASCAL" ≤ "pascal") OU NON ("P" ≠ "p")
- NON (13 MOD 2 = 0) OU (1+1 = 3)

Leçon 3

Le type énuméré et le type intervalle

I – Le type scalaire énuméré

Exemples :

mois = (Janvier, février, mars, Avril, Mai, juin, juillet, août, septembre, octobre, novembre, décembre)

Couleurs = (Rouge, Bleu, Jaune)

a. Définition

Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs désignées par des identificateurs. Un type énuméré est constitué d'un nombre limité de valeurs. (Maximum 256 en Pascal)

b. Déclaration

Type
nom_type = (val1, val2,, valn)

Objet	Type	Rôle
Nom_variable	nom_type	Rôle

En analyse

Type **nom_type** = (val1, val2,, valn);

Var nom_variable : **nom_type**;

En Pascal

c. Opérateurs

- Relationnels : = <= >= <> < >
- Exemple : janvier < mars (par rapport au type **mois**).
- L'affectation
- Les opérations d'entrée / sortie ne sont pas possibles avec les types énumérés.

d. Fonctions prédéfinies sur les types énumérés

Remarque : A chaque valeur énumérée correspond un numéro d'ordre. La numérotation commence à partir de 0.

- Succ** ()
- Pred** ()
- Ord** ()

II – Le type intervalle

Le type intervalle permet de limiter les valeurs d'un type scalaire discret comme le type entier, caractère ou énuméré. L'intervalle s'exprime au moyen de valeurs limites selon la forme suivante :

Binf .. Bsup

Exemples :

Type

age = 0 .. 150;

{par rapport au type **entier**}

majuscule = 'A' .. 'Z';

{par rapport au type **caractère**}

trimestre_1 = janvier .. mars ;

{par rapport au type **mois**}

- **Application :**

Créer un programme en pascal pour créer et tester les types mois et trimestre1.

Leçon 4

Les tableaux à une dimension

I – Déclaration d'un tableau

1. **Définition** (voir livre page 37)
2. **Caractéristiques**
 - Identificateur
 - Nombre d'éléments (cases)
 - Type de ses éléments
3. **Déclaration d'un tableau** : (livre page 37)

Un tableau unidimensionnel est appelé aussi vecteur.

4. **Accès à l'ⁱ^{ème} élément d'un tableau T d'entiers** :

En analyse :

Saisie : T [i] = donnée

Affectation : T [i] ← 20

Affichage : écrire (T [i])

En Pascal :

Saisie : readln (T [i])

Affectation : T [i] := 20

Affichage : writeln (T [i]);

II – Type tableau

1. **Intérêt** : On créant un type tableau, on peut déclarer par la suite autant d'objet tableau de ce type.
2. **Déclaration** : (livre page 40)