Algorithmique
et
Programmation

3ème de l'enseignement Secondaire
Section : **Sciences de l'Informatique**

Professeur
Mohamed TRABELSI

Plan du Cours

Rappel

3 SI

Exercice de révision

Objectif : rappeler la démarche de résolution adoptée pour résoudre un problème et aboutir à un programme. Application page 10 livre algorithmique 3ème SI.

Enoncé :

On se propose de calculer l'allongement L d'un ressort de raideur K auquel est accroché une masse m. Sachant que m * g = K * L.

- Présenter la spécification de ce problème
- Déduisez l'algorithme
- Traduisez la solution en Pascal et l'exécutez pour m = 150 et K = 10.

a. Analyse

Résultat : Afficher l'allongement L
Traitement :
L ← m * **g** / K
m = donnée
K = donnée
g est une constante de valeur 9,8

b. Algorithme

0) Début allongement
1) Lire (K)
2) Lire (m)
3) L ← m * **g** / K
4) Ecrire (L)
5) Fin allongement

T.D.O

Objet	Type	Rôle
K, m, l	Réel	K : raideur, m : masse, l : allongement
g	Constante = 9,8	pesanteur

c. Pascal (voir fichier : allonge.pas)

Version 2 : Tester la valeur de K qui doit être > 0. Deux solutions

- Faire une saisie contrôlée de K.
- Tester la valeur K avant d'effectuer le calcul.

Algorithme

0) Début allongement
1) Répéter
   écrire (" Donner la valeur de la raideur du ressort : "), lire (K)
   Jusqu'à K > 0
2) écrire (" Donner la valeur de la masse : "), lire (m)
3) L ← m * **g** / K
4) Ecrire (L)
5) Fin allongement

Chapitre 1

Structures de données et structures simples

Durée : 12 Heures
Type : Théorique / Pratique

I. Les constantes et les variables

I.1. Les constantes

- Une constante est caractérisée par un nom et une valeur inchangeable.
- Exemple voir exercice révision (calcul allongement).
- Déclaration :
  - Côté analyse : **nom_contante** est une constante de valeur **valeur_constante**
  - Côté T.D.O :

Objet	Type	Rôle
nom_contante	Constante = **valeur_constante**	

  - Côté Pascal :

CONST nom_contante **= valeur_constante;**

I.2. Les variables

Une variable est une zone mémoire réservée pour contenir une valeur d'un type bien déterminé. Cette valeur pourra changer tout au long du programme.

- **Caractéristiques :**

  Voir livre page 11.

- **Application 1 :**

  Calculer et afficher la distance entre deux points M et N dont les coordonnées sont des entiers données. (Par rapport à un repère gradué)

  M (a, b) et N (c, d)

a. Analyse

Résultat : Afficher la distance DT

Traitement :

DT ← **SQRT (**SQR (a – c) + SQR (b – d) **)**

(a, b) = Données

(c, d) = Données

b. Algorithme

0) Début distance
1) Lire (a, b)
2) Lire (c, d)
3) DT ← **SQRT (**SQR (a – c) + SQR (b – d) **)**
4) écrire (DT)
5) Fin distance

T.D.O

Objet	Type	Rôle
a, b	entier	Coordonnées de M
c, d	entier	Coordonnées de N
DT	Réel	Distance entre M et N

c. Pascal (voir fichier : distance.pas)

II. Les types standard

II.1 Le type entier

- Le type entier permet de manipuler des valeurs de l'ensemble **Z**.
- **Domaine de définition et opérateurs :** Voir tableau page 14**.**

Activité 1 :

Déclarer une variable qui servira à stocker l'âge d'une personne.

Déclarer une variable qui servira à stocker des factoriels avec 1 < n < 20.

Quels sont les types de données que vous allez choisir pour ces 2 variables ?

- **Ouvrir Pascal et voir le help (clic droit sur integer)**

- **Sous types du type entier**
  - *Non signés* **:**

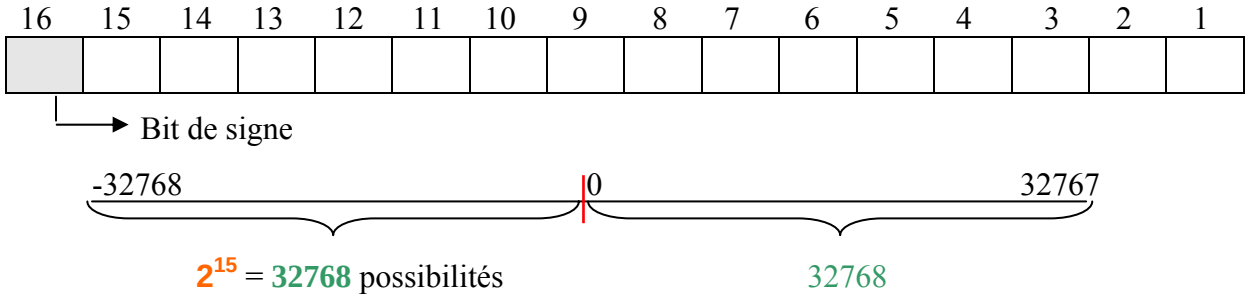
Type	**En pascal**	**Domaine de valeur**	**Taille**
Octet	byte	0 → 255	1 octet
Mot	word	0 → 65535	2 octets

  - *Signés* **:**

Type	**En pascal**	**Domaine de valeur**	**Taille**
Entier court	shortint	-128 → 127	1 octet
Entier	Integer	-32768 → 32767	2 octets
Entier long	Longint	-2147483648 .. 2147483647	4 octets

- **Représentation d'une valeur de type Entier en mémoire centrale :**

Elle s'effectue sur 2 octet ⇔ 16 bits avec un bit réservé au signe 1 si < 0 ; 0 si >= 0.



II.2 Le type réel (real)

En stockant des nombres réels dans des structures de données on se retrouve face à 2 limites :

1. La limite de grandeur
2. La limite de la précision (nombre de chiffre après la virgule)

Le traitement et l'affichage des données de ce type se font en virgule flottante, dans une notation dite scientifique à base 10.

- **Domaine de définition et opérateurs :** Voir tableau page 17**.**

Activité 2 :

Ecrire un programme en Pascal qui permet de saisir un réel et de le réafficher.
Procéder à la série de tests suivants :

- Valeur 1 : 5.223
- Valeur 2 : 13251.33
- Valeur 3 : 0.0999

Que pouvez vous en déduire ?
Voir dans le help les sous types du type réel

- **Les fonctions arithmétiques standard :** voir livre page 19**.**

Activité 3 :

Ecrire un programme en Pascal qui permet de saisir un réel *x* et d'afficher le résultat de toutes les fonctions arithmétiques standard.

II.3 Le type booléen (boolean)

- **Domaine de définition et opérateurs logiques :** voir tableau page 21**.**

Activité 4 :

- Faire l'activité 12, page 21
- Evaluez l'expression logique suivante sachant que : a = 4, b = 5, c = 1 et d = 0

  (a < b) OU (c – d = a) ET Non (a ≠ b)
- Règles de priorité entre les opérateurs : Non, ET, OU et OUex

II.4 Le type caractère (char)

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)

- **Domaine de définition et opérateurs relationnels :** Voir tableau page 23**.**
- **Les fonctions prédéfinies sur les caractères :** Voir livre page 23**.**

Activité 5 :

Ecrire un programme en Pascal qui permet de saisir un caractère *c* et de changer sa casse.
Exemple : c = "A" → "a"
c = "a" → "A"

a. Analyse

Résultat : Afficher le caractère *c* modifié
Traitement :

```
Si c  est majuscule alors
                    c ← Chr (Ord (c) + 32)
          sinon
                    c ← Upcase (c)
Fin si
c = donnée ("Donner un caractère : ")
```

b. Algorithme

```
0) Début min_maj
1) Ecrire ("Donner un caractère : "), (Lire (c)
2) Si c  dans ["A".."Z"] alors
                    c ← Chr (Ord (c) + 32)
          Sinon
                    c ← Upcase (c)
   Fin si
3) écrire (c)
4) Fin min_maj
```

c. Pascal (voir fichier : min_maj.pas)

II.5 Le type chaîne de caractères (string)

Une variable chaîne de caractère est une suite de n caractère, n compris entre 0 et 255.

On peut accéder au $i^{ème}$ caractère d'une chaîne CH grâce à la notation CH[i].

- **Déclaration :** Lors de la déclaration d'une chaîne on peut préciser la taille voulue.

Objet	Type	Rôle
nom_chaîne	Chaîne [longeur_chaîne]	

- **Affectation :**
  - Côté analyse et algorithmique : exemple CH ← "Sciences informatiques"
  - Côté Pascal : exemple CH := 'Sciences informatiques'
- **Les fonctions et les procédures prédéfinies sur les chaînes :** Voir livre page 26**.**
- Activité 14 page 27 et applications**.**

III. Les structures simples

- **Définition :** Voir livre page 28**.**

III.1 L'opération d'entrée (La saisie)

a. En analyse :

Objet = donnée ("commentaire sur la variable")

b. En algorithmique :

```
Ecrire ("commentaire sur la variable")
Lire (Objet)
```

c. En Pascal :

```
writeln (' commentaire sur la variable');
readln (Objet);
```

III.2 L'opération de sortie

En analyse et algorithmique :

```
Ecrire (Objet)
Ecrire ("message ", Objet)
Ecrire ("message 1 ", Objet 1, "message 2 ", Objet 2)
```

En Pascal :

```
writeln (Objet);
writeln ('message' , Objet);
writeln ('msg 1 ' , Objet 1, 'msg 2 ' , Objet 2);
```

III.3 L'opération d'affectation

En analyse et algorithmique :
←

En Pascal :
:=

Rappel sur la notion de **permutation** :
Avec variable intermédiaire
Sans variable intermédiaire
Voir livre page 31

- **L'affectation :**
  Soit la séquence d'affectation suivante :

  1) R ← 10
  2) S ← pi * R * R
  3) R ← 1
  4) S ← carré (ABS(R))

  Remplir le tableau d'exécution de cette séquence d'affectation.

	Valeur des variables	
N° instruction	**R**	**S**
1		
2		
3		
4		

IV. Les types utilisateurs

IV.1 Les types énumérés

Exemples :

Jours_de_la_semaine = (Dimanche, Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi)

Couleurs = (Rouge, Bleu, Jaune)

a. Définition

Les types énumérés permettent de représenter des valeurs en les énumérant au moyen de leurs noms. Un type énuméré est constitué d'un nombre limité de valeurs. (Maximum 256 en Pascal). Ces valeurs ne doivent pas appartenir à un autre type de données.

b. Déclaration

Type
nom_type = (val1, val2,……, valn)

Objet	**Type**	**Rôle**
nom_variable	**nom_type**	Rôle

En algorithmique

Type nom_type = (val1, val2,……, valn);
Var nom_variable : **nom_type;**

En Pascal

c. Opérateurs relationnels

= <= >= <> < >

Exemple : Lundi < Mardi (par rapport au type jours_de_la_semaine).

d. Les fonctions prédéfinies sur les types énumérés

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

- **Voir livre page 37**
- **Voir fichier** (enumere1.pas) – démonstration.

IV.2 Le type intervalle

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

Binf .. Bsup

Exemples :

```
Type
     age = 0 .. 150;

     majuscule = 'A' .. 'Z';
Var
     JDT = Lundi .. Vendredi ;
```

{par rapport au type **byte**}

{par rapport au type **char**}

{Déclaration du type directement dans la partie var par rapport au type **jours de la semaine**}

V. Les tableaux

V.1 Les tableaux à une dimension (vecteurs)

Activité 6 :

Faire l'analyse d'un programme qui permet de calculer la paye hebdomadaire d'un ouvrier.

- La solution doit nous permettre de :
  - Saisir le nombre d'heures travaillées par jour. (Jours ouvrables : Du lundi au Vendredi)
  - Saisir la valeur du taux horaire de payement
- Puis de calculer le nombre total d'heures travaillées de la semaine et d'afficher le montant de la paye.

a. Définition (voir livre page 40)

b. Déclaration

Pour déclarer un tableau on peut procéder de deux manières :

- Créer directement le tableau dans la zone de déclaration des variables.
- Créer un type tableau puis un tableau de ce type.

(Voir livre page 40, 41)

c. Correction de l'activité

Version 1

1. *Analyse*

Résultat : écrire ("La paye de la semaine est: ", MP, " DT")

Traitements :

MP ← total * THP

Pour saisir le tableau des horaires H et au même temps calculer la somme des horaires stockés dedans il faut :

i. Initialiser total à 0

ii. Utiliser une boucle Pour qui permet de saisir la case H[i] puis de cumuler cette valeur dans S.

THP = donnée ("Précisez la valeur du taux horaire : ")

2. *Algorithme*

0) Début paye_semaine
1) total ← 0

   Pour i de 1 à 5 faire

   écrire ("Entrez le nombre d'heure du jour ", i, " : "), lire (H[i])
   total ← total + H[i]

   Fin pour
2) Ecrire ("Précisez la valeur du taux horaire : "), lire (THP)
3) MP ← total * THP
4) écrire ("La paye de la semaine est: ", MP, " DT")
5) Fin paye_semaine

Type
Paye = tableau de 5 octet

T.D.O

Objet	*Type*	*Rôle*
i	octet	Compteur
total	octet	Total heures de la semaine
THP	Réel	Taux horaire de payement
MP	Réel	Paye de la semaine
H	paye	Tableau des heures travaillées par jour

3. *Pascal* (voir fichier : paye_sm1.pas)

Version 2

Algorithme

0) Début paye_semaine2
1) total ← 0

   Pour j de **lundi** à **vendredi** faire

      écrire ("Entrez le nombre d'heure du jour ", ord(j) +1, " : "), lire (H[j])

      total ← total + H[j]

   Fin pour
2) Ecrire ("Précisez la valeur du taux horaire : "), lire (THP)
3) MP ← total * THP
4) écrire ("La paye de la semaine est : ", MP, " DT")
5) Fin paye_semaine2

Type
jours = (dimanche, lundi, mardi, mercredi, jeudi, vendredi, samedi) jours_de_travail = lundi ..vendredi Paye = tableau de 5 octet

T.D.O

Objet	Type	Rôle
j	jour_de_travail	Représente le jour en cours
total	octet	Total heures de la semaine
THP	Réel	Taux horaire de payement
MP	Réel	Paye de la semaine
H	paye	Tableau des heures travaillées par jour

a. Pascal (voir fichier : paye_sm2.pas)

V.2 Les tableaux à deux dimensions

Activité 7 :

Faire l'analyse d'un programme qui permet de calculer la paye hebdomadaire de n ouvrier sachant que (1≤ n ≤ 5).

- La solution doit nous permettre de saisir :
  - Le nombre d'ouvrier **(n)**
  - Le nombre d'heures travaillées par jour.
  - La valeur du taux horaire de payement
- Puis de calculer le nombre total d'heures travaillées de la semaine et d'afficher **simultanément** le montant de la paye pour chaque ouvrier.

a. Définition

Un tableau à deux dimensions a la forme d'une matrice composée de lignes et de colonnes. Toutes les cases de ce tableau auront le même type. L'accès à une case de ce tableau s'effectue par conséquent grâce à deux indices comme suit :

M [i , j] où i représente l'indice de la ligne et j celui de la colonne.

b. Déclaration

- Création d'un type tableau à deux dimensions.

Type
nom_type = tableau de [**intervale1** , **intervale2**] de Type

Objet	**Type**	**Rôle**
nom_tableau	**nom_type**	Rôle

En algorithmique

Type nom_type = array [**intervale1**, **intervale2**] of Type;
Var nom_ tableau : **nom_type;**

En Pascal

Activité 8 :

Faire l'analyse d'un programme qui permet de saisir les températures de n villes pendant une semaine. (1≤ n ≤ 10)

- Afficher ce tableau tout en précisant la moyenne des températures par ville.
- *Afficher aussi l'écart type des températures par ville.*

Exemple :

Villes	Dim	Lun	Mar	Mer	Jeu	Ven	Sam	Moyenne	Ecart type
Tunis	20	23	21	22	26	27	23	23,14	...
Gabes	22	23	24	24	25	28	27	24,71	...

Moyenne = $(\sum_{i=1}^{n} Mi) / n$ écart type = $\sqrt{\sum_{i=1}^{n} (Mi - Moy)^2}$

a. Analyse du problème

Résultat : Affichage

Traitements :

- *Prévoir deux boucles imbriquées ayant comme indice respectif i et j permettant d'afficher les températures de la matrice MT et leurs moyennes par ville à partir du tableau Moyenne.*
- *Prévoir deux boucles imbriquées ayant comme indice respectif i et j permettant de saisir les températures de la matrice MT, de calculer au fur et à mesure le cumul des températures par villes, de calculer la moyenne et de la stocker dans le tableau Moyenne.*

  n = Donnée ("Donner le nombre de ville : ")

b. Algorithme

```
0) Début temperature_ville
1) écrire ("Donner le nombre de ville : "), lire (n)
2) Pour i de 1 à n faire
        cumul ← 0
        Pour j de 1 à 7 faire
            écrire ("Donner la température : "), lire (MT [ i , j ])
            cumul ← cumul + MT [ i , j ]
        fin pour
        moyenne [i] ← cumul / 7
   Fin pour
3) Pour i de 1 à n faire
        Pour j de 1 à 7 répéter
            Ecrire (MT [ i , j ], "  ")
        Fin pour
        Ecrire (moyenne [i])
   Fin pour
4) Fin temperature_ville
```

Type
T_moyenne = tableau de 10 réels T_température = tableau [1..10, 1..7] de réels

T.D.O

Objet	Type	Rôle
i, j	octet	Compteurs
n	octet	Nombre des villes
MT	T_température	Matrice des températures
moyenne	T_moyenne	Tableau des moyennes des températures
cumul	réel	La somme des températures

c. Pascal (voir fichier : tempvile.pas)

a. Analyse (*Version 2)*

Résultat : Affichage

Traitements :

- *Prévoir deux boucles imbriquées ayant comme indice respectif i et j permettant de saisir les températures de la matrice MT, de calculer au fur et à mesure le cumul des températures par villes, de calculer la moyenne et de l'afficher sur écran.*

  n = Donnée ("Donner le nombre de ville : ")

b. Algorithme

0) Début temperature_ville2
1) écrire ("Donner le nombre de ville : "), lire (n)
2) Pour i de 1 à n faire
   cumul ← 0
   Pour j de 1 à 7 faire
   écrire ("Donner la température : "), lire (MT [i , j])
   cumul ← cumul + MT [i , j]
   fin pour
   écrire (cumul / 7)
   Fin pour
3) Fin temperature_ville2

Type
T_température = tableau [1..10, 1..7] de réels

T.D.O

Objet	Type	Rôle
i, j	octet	Compteurs
n	octet	Nombre des villes
MT	T_température	Matrice des températures
cumul	réel	La somme des températures