

OFPPT

SECTEUR NTIC

**Filières : TDI / TDM / TRI**

*Niveau Technicien spécialisé*

**Manuel de TP**

**Module 7 : Techniques de Programmation Structurée**

2015

## **Conception et rédaction**

<b>NOM et PRENOM</b>	<b>AFFECTATION</b>
<b>BOUROUS Imane</b>	<b>ISTA HAY RIAD - Rabat</b>

**Liste des équipements: Néant**

**Liste de matière d'œuvre et outillage : Néant**

**Nous tenons à remercier les formateurs concepteurs des Manuels de TP et nous invitons les utilisateurs de ce manuel à communiquer, par email, à la direction du CDC TIC toutes les remarques et suggestions afin de les prendre en considération pour l'enrichissement et l'amélioration de ce document.**

## MODULE 07 : TECHNIQUES DE PROGRAMMATION STRUCTURÉE

Code : TRI-07

Durée : 120 h

PRÉCISIONS	ÉLÉMENTS DE CONTENU
1. Comprendre la notion de résolution de problèmes par ordinateur. 2. Connaître le principe <i>Entrée de données – Traitement – Sortie des résultats</i> .	<ul style="list-style-type: none"> <li>• Méthode de résolution de problème par ordinateur :               <ul style="list-style-type: none"> <li>✓ conception par le développeur ;</li> <li>✓ exécution par l'ordinateur ;</li> <li>✓ exploitation par l'utilisateur.</li> </ul> </li> <li>• Structure d'un organigramme hiérarchique (OH).</li> <li>• Technique de construction des OH.</li> <li>• Données d'entrée et de sortie.</li> <li>• Nature des traitements.</li> <li>• Conditions d'exécution de l'algorithme.</li> </ul>
<b>A. Analyser les besoins en ce qui a trait au développement de l'application.</b>	<ul style="list-style-type: none"> <li>• Analyse du besoin.</li> <li>• Description du contexte et reformulation des objectifs.</li> <li>• Identification de la nature des objets manipulés.</li> <li>• Distinction des entrées, des sorties et des traitements.</li> <li>• Détection des contraintes liées au développement de l'application.</li> </ul>
3. Comprendre l'utilité d'un algorithme. 4. Connaître les différents types de représentation existants : pseudo-code, organigramme, etc. 5. Connaître les notions de base de l'algèbre de Boole.	<ul style="list-style-type: none"> <li>• Structure d'un algorithme et normes de représentation.</li> <li>• Rappel de l'algèbre de Boole, valeurs de vérité et tables de vérité des opérateurs logiques : ET, OU, NON.</li> <li>• Règles de décomposition des traitements et stratégies d'éclatements.</li> </ul>
<b>B. Utiliser les instructions de base de l'algorithme.</b>	<ul style="list-style-type: none"> <li>• Instructions de base d'un algorithme :               <ul style="list-style-type: none"> <li>✓ lecture ;</li> <li>✓ écriture ;</li> <li>✓ affectation.</li> </ul> </li> <li>• Rôle du typage des données :               <ul style="list-style-type: none"> <li>✓ portée ;</li> <li>✓ champs des valeurs ;</li> <li>✓ opérateurs).</li> </ul> </li> <li>• Instructions alternatives et répétitives.</li> </ul>
6. Comprendre l'utilité de l'optimisation du programme.	<ul style="list-style-type: none"> <li>• Intérêt de la réutilisation du code.</li> </ul>
<b>C. Optimiser l'algorithme.</b>	<ul style="list-style-type: none"> <li>• La portée des variables.               <ul style="list-style-type: none"> <li>✓ variables locales ;</li> <li>✓ variables globales ;</li> <li>✓ variables statiques.</li> </ul> </li> <li>• Les procédures :               <ul style="list-style-type: none"> <li>✓ passage de paramètres par valeur ;</li> <li>✓ passage de paramètres par référence.</li> </ul> </li> <li>• Les fonctions.</li> <li>• Exploitation des fonctions et procédures dans un programme principal.</li> </ul>

PRÉCISIONS	ÉLÉMENTS DE CONTENU
<b>D. Utiliser des structures de données complexes.</b>	<ul style="list-style-type: none"> <li>• Les types complexes : <ul style="list-style-type: none"> <li>✓ les tableaux ;</li> <li>✓ les chaînes de caractères;</li> <li>✓ les enregistrements ;</li> <li>✓ les tableaux d'enregistrements.</li> </ul> </li> <li>• Les algorithmes de tri.</li> <li>• Les algorithmes de recherche rapide.</li> </ul>
7. Comprendre l'intérêt de sauvegarde de données. 8. Comprendre la structure et l'organisation d'un disque. 9. Connaître les différents types d'organisation de fichiers. 10. Présenter les principales normes de codage de caractères 11. Connaître les différentes méthodes d'accès	<ul style="list-style-type: none"> <li>• Intérêt de sauvegarde de données sous forme de fichiers.</li> <li>• Les notions de fichier d'échange entre mémoire et périphérique de stockage</li> <li>• Les options d'ouverture, de partage et les opérations sur les enregistrements</li> <li>• Les différents types de fichiers : <ul style="list-style-type: none"> <li>✓ les fichiers binaires ;</li> <li>✓ les fichiers texte ASCII et Unicode.</li> </ul> </li> <li>• Les différents types d'accès <ul style="list-style-type: none"> <li>✓ séquentiel ;</li> <li>✓ direct.</li> </ul> </li> </ul>
<b>E. Manipuler les fichiers de données.</b>	<ul style="list-style-type: none"> <li>• Déclaration des fichiers : <ul style="list-style-type: none"> <li>✓ chemin d'accès ;</li> <li>✓ mode d'ouverture (lecture, écriture, lecture/écriture) ;</li> </ul> </li> <li>• Manipulation des fichiers : <ul style="list-style-type: none"> <li>✓ Lecture ;</li> <li>✓ écriture ;</li> <li>✓ recherche ;</li> <li>✓ modification ;</li> <li>✓ suppression ;</li> </ul> </li> </ul>

## Activité d'apprentissage 1

Module	TRI-07	Techniques de programmation structurée
Précision	A	Analyser les besoins en ce qui a trait au développement de l'application
Code Activité	A001	
Activité	Réaliser l'analyse et la construction logique d'un programme	
Durée	2 heures	
Phase d'apprentissage	BASE	

Détails sur les objectifs visés par l'activité	Cette activité d'apprentissage doit vous permettre de :
	<ul style="list-style-type: none"><li>▪ Analyser les besoins et préciser l'objectif attendu du programme.</li><li>▪ Déterminer les données en entrée, en sortie et les traitements nécessaires</li><li>▪ Présenter la logique du programme sous la forme d'une représentation hiérarchique</li></ul>

Matière d'œuvre et/ou outillage	.....
	.....
	.....

## DESCRIPTION DE L'ACTIVITÉ



La responsable de la paie, Madame Maliki, souhaite connaître le montant net du salaire des employés de la boutique « Malabis ». Le salaire net résulte du montant brut du salaire auquel sont retranchés les montants de charge sociale de la CNSS, une partie pour la retraite et une partie pour la santé, et le montant de l'impôt sur le revenu qui est retenu à la source.

Pour les charges sociales le calcul s'effectue selon les modalités suivantes :

- Si le salaire est inférieur ou égal à 12 000 MAD, CNSS Santé est de 3,54 % et CNSS Vieillesse 2,56 %
- Si le salaire est supérieur à 12 000 MAD, CNSS Santé est majorée de 0,86 % et CNSS Vieillesse de 0,8 % pour la tranche de salaire excédant 12 000 MAD

Pour l'impôt le calcul s'effectue selon les modalités suivantes :

- Si le salaire est inférieur ou égal à 10 000 MAD, le pourcentage retenu est de 20,56 %
- Si le salaire est supérieur à 10 000 MAD, la majoration est de 10 % pour la tranche de salaire excédant 10 000 MAD

Dans un premier temps, le programme ne traite que d'un salarié à la fois

Travail à réaliser :

Déterminer :

- L'objectif
- Les données en entrée
- Les données en sortie
- Les traitements à appliquer sur les données en entrée pour obtenir le résultat

Réaliser l'ordinogramme du programme. Les données en entrée sont acquises dans le cadre d'un dialogue utilisateur.

## Activité d'apprentissage 2

Module	TRI 07	Techniques de programmation structurée
Précision	A	Analyser les besoins en ce qui a trait au développement de l'application
Code Activité	A002	
Activité	Réaliser l'analyse et la construction logique d'un programme	
Durée	2 heures	
Phase d'apprentissage	ENTRAINEMENT	

Détails sur les objectifs visés par l'activité	Cette activité d'apprentissage doit vous permettre de :
	<ul style="list-style-type: none"><li>▪ Analyser les besoins et préciser objectif attendu du programme.</li><li>▪ Optimiser les traitements et les fonctionnalités d'un programme</li><li>▪ Déterminer un jeu d'essai judicieux pour vérifier le bon fonctionnement du programme</li><li>▪ Présenter la logique du programme sous la forme d'une représentation hiérarchique</li></ul>

Matière d'œuvre et/ou outillage	.....
	.....
	.....

---

## DESCRIPTION DE L'ACTIVITÉ



Sur la base de l'exercice précédent.

Madame Maliki souhaite ne plus être obligée de lancer le programme pour chaque salarié. Elle indique qu'en général, le calcul se fait en fin de mois pour l'ensemble des salariés. Elle a oublié de vous préciser lors de l'expression de son besoin initial qu'il lui fallait aussi connaître les différents montants de charges sociales et d'impôts qu'elle devaient reporter sur le document destiné à la CNSS et au Ministère des Finances.

Travail à réaliser :

Déterminer :

- Les objectifs complémentaires
- Les données en entrée supplémentaires
- Les données en sortie supplémentaires
- Les nouveaux traitements à mettre en œuvre

Réalisez l'ordinogramme du programme. Les données en entrée sont acquises dans le cadre d'un dialogue utilisateur.

Proposez un jeu d'essai pour tester le bon fonctionnement du programme.

Réfléchissez à la justification des valeurs choisies pour le jeu d'essai.



### Activité d'apprentissage 3

Module	TRI-07	Techniques de programmation structurée
Précision	B	Utiliser les instructions de base de l'algorithme.
Code Activité	B001	
Activité	Instructions de bases : lecture, écriture et affectation	
Durée	2 heures	
Phase d'apprentissage	BASE	

Détails sur les objectifs visés par l'activité	Cette activité d'apprentissage doit vous permettre de :
	Connaître les Instructions de base d'un algorithme : <ul style="list-style-type: none"><li>✓ lecture ;</li><li>✓ écriture ;</li><li>✓ affectation.</li></ul> Rôle du typage des données <ul style="list-style-type: none"><li>✓ portée ;</li><li>✓ champs des valeurs ;</li><li>✓ opérateurs).</li></ul>

Matière d'œuvre  
et/ou outillage

.....

.....

.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui permet d'échanger les valeurs de deux entiers A et B (les valeurs de A et B, vont être saisies par l'utilisateur). Il est demandé dans ce programme :

- D'afficher les deux valeurs initiales de A et B.
- Effectuer la permutation de A et B.
- Afficher les deux valeurs de A et B permutées.

**Exercice 2 :** Écrire un algorithme qui permet de faire la somme, la soustraction, la multiplication et enfin la division de deux nombres réels X et Y. Afficher tous les résultats.

**Exercice 3 :** Écrire un algorithme qui lit le prix unitaire HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

**Exercice 4 :** Un magasin dispose de cinq produits :

Produit A : prix 5.00 DH

Produit B : prix 2.50 DH

Produit C : prix 3.00 DH

Produit D : prix 10.00 DH

Produit E : prix 7.00 DH

Un client achète :

X unités du produit A, Y unités du produit B, Z unités du produit C, T unités du produit D, U unités du produit E.

On désire calculer et afficher :

Le prix hors taxe (PHT) de cette vente.

La taxe sur la valeur ajoutée (TVA)

Le prix toutes taxes comprises (PTTC) de cette vente

On donne le taux de TVA :  $TTVA=0.20$

1- Analyser le problème.

2- Écrire un algorithme qui permet de calculer et afficher :

Le prix hors taxe, la taxe sur la valeur ajoutée et le prix toutes taxes comprises ?

(L'utilisateur doit entrer le nombre d'articles achetés de chaque produit)

## Activité d'apprentissage 4

Module TRI-07 Techniques de programmation structurée

Précision B Utiliser les instructions de base de l'algorithme.

Code Activité B002

Activité Instructions alternatives.

Durée 5 heures

Phase d'apprentissage BASE

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Déterminer les instructions et la logique d'un programme simple à l'aide d'un pseudo langage.
- Exprimer correctement des conditions simples
- Exprimer correctement des conditions complexes imbriquées

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

**Exercice 2 :** Écrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

**Exercice 3 :** Écrire un algorithme qui permet la résolution d'une équation du premier degré (une équation sous la forme  $ax+b=0$ )

**Exercice 4 :** Écrire un algorithme qui permet la résolution d'une équation du second degré (une équation sous la forme  $ax^2+bx+c=0$ )

**Exercice 5 :** Ecrire un algorithme qui range trois nombres donné x, y, z, dans l'ordre croissant ( $x < y < z$ ).

**Exercice 6 :** "Calculer le lendemain d'une journée donnée (jour, mois, année)"  
On ne tiendra pas compte ici des années bissextiles, le mois de février aura toujours 28 jours.

**Exercice 7 :** "Calculer la durée d'un trajet connaissant l'heure de départ et d'arrivée". On se contente des heures et des minutes, la durée totale ne dépassera jamais 24 heures.

**Exercice 8 :** Écrire un algorithme qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de la somme de A et B sans faire l'addition.

## Activité d'apprentissage 5

Module TRI-07 Techniques de programmation structurée

Précision B Utiliser les instructions de base de l'algorithme.

Code Activité B003

Activité Instructions alternatives.

Durée 3 heures

Phase d'apprentissage ENTRAINEMENT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Déterminer les instructions et la logique d'un programme simple à l'aide d'un pseudo langage.
- Exprimer correctement des conditions complexes imbriquées

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 1:

La direction d'un supermarché a décidé d'accorder des réductions à ses clients selon le montant d'achat.

La réduction est calculée selon les règles suivantes:

- 20% pour un montant d'achat de plus de 5000 dhs
- 15% pour un montant d'achat entre 3000 dhs < montant d'achat ≤ 5000 dhs
- 10% pour un montant d'achat entre 1000 dhs < montant d'achat ≤ 3000 dhs
- Aucune réduction pour un montant d'achat inférieur à 1000 dhs.

Ecrire un algorithme qui permet de calculer et d'afficher la réduction et montant à payer

### Exercice 2:

•La direction d'une entreprise désire automatiser le calcul de l'indemnité à verser aux cadres en cas de licenciement.

•Après d'ancienneté, dans l'entreprise, il sera alloué aux cadres licenciés une indemnité tenant compte de leur ancienneté et s'établissant comme suit:

•La moitié du salaire d'un mois par année d'ancienneté : pour la tranche d'ancienneté entre 1 an et 10 ans.

•Au-delà de 10 ans un mois de salaire par année d'ancienneté.

•Une indemnité supplémentaire serait allouée aux cadres âgés de plus de 45 ans de:

- 2 mois de salaire si le cadre est âgé de 46 à 49 ans.
- 5 mois si le cadre est âgé de plus de 50 ans.

Ecrire un algorithme qui permet de saisir l'âge, l'ancienneté et le dernier salaire et d'afficher l'indemnité du cadre.

### Exercice 3 :

Une papeterie facture 0,50 dhs les dix premières photocopies, 0,30 dhs E les vingt suivantes et 0,25 dhs au-delà. Ecrire un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

## Activité d'apprentissage 6

Module TRI-07 Techniques de programmation structurée

Précision B Utiliser les instructions de base de l'algorithme.

Code Activité B004

Activité Instructions Répétitives.

Durée 4 heures

Phase d'apprentissage BASE

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Déterminer les instructions et la logique d'un programme simple à l'aide d'un pseudo langage.
- Exprimer des itérations
- Construire des dialogues utilisateur pertinents

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 5) :

**Table de 5 :**

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

...

$$5 \times 10 = 50$$

**Exercice 2 :** Écrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

$$1 + 2 + 3 + 4 + 5 = 15.$$

**Exercice 3 :** Écrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

**NB :** la factorielle de 8, notée 8 ! vaut  $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$

**Exercice 4 :** Écrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 10 nombres et sa position :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

...

Entrez le nombre numéro 10 : 6

Le plus grand de ces nombres est : 14, sa position : 2

**Exercice 5 :** Écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

**Exercice 6 :** Écrire un algorithme qui demande successivement des nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces nombres et quel était sa position. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

**Exercice 7 :** Écrire un algorithme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.



## Activité d'apprentissage 7

Module TRI-07 Techniques de programmation structurée

Précision B Utiliser les instructions de base de l'algorithme.

Code Activité B005

Activité Instructions Répétitives.

Durée 9 heures

Phase d'apprentissage ENTRAÎNEMENT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Déterminer les instructions et la logique d'un programme simple à l'aide d'un pseudo langage.
- Exprimer des itérations
- Construire des dialogues utilisateur pertinents

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui permet de calculer la somme :  
 $1 - 1/2 + 1/2^2 - 1/2^3 + 1/2^4 - \dots + 1/2^n$  (^ : désigne la puissance)

**Exercice 2 :** Écrire un algorithme qui vérifie si un nombre entier saisi par l'utilisateur est un nombre premier ou pas.

Un nombre premier est un nombre qui ne permet la division que sur 1 et lui-même.

Exemples de nombre premier : 2, 5, 7, 11, 13, 31, ...

**Exercice 3:** Écrire un algorithme qui permet de calculer la somme de la suite suivante

$S = 1 + 2 + 4 + 7 + 11 + 16 + 22 + 29 + \dots$

**Exercice 4:** Écrire un algorithme qui demande à l'utilisateur de saisir une somme d'argent, puis affiche la décomposition de cette somme d'argent en billets et pièces de (200, 100, 50, 20, 10, 5, 2 et 1 dh) en utilisant le moins de billets et de pièces possibles.

**Exercice 5:** On considère la suite :  $u(0) = a$  (a entier)

si  $u(n)$  pair alors  $u(n+1) = u(n)/2$  sinon  $u(n+1) = 3 * u(n) + 1$

Pour toutes les valeurs a, il existe un entier N tel que  $u(N) = 1$

Écrire un algorithme qui demande à l'utilisateur de taper a et qui affiche toutes les valeurs de  $u(n)$  de  $n=1$  à  $n=N$ .

**Exercice 6:** Écrire un algorithme qui demande à l'utilisateur de saisir deux nombres a et b (tel que  $a < b$ ) et affiche tous les nombres premiers compris entre ces deux nombres.

**Exercice 7:** Écrire un algorithme permettant d'afficher le triangle suivant : (le nombre de ligne est saisi par l'utilisateur)

```
1
12
123
1234
12345
123456
1234567
...
```

**Exercice 8:** Écrire un algorithme qui calcule le terme  $U_N$  de la suite de Fibonacci donnée par :  $U_1 = 1, U_2 = 1, U_N = U_{N-1} + U_{N-2}$  (pour  $N > 2$ )

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 9 :** Calculez le nombre lu à rebours d'un nombre positif entré au clavier en supposant que le fichier d'entrée standard contient une suite de chiffres non nuls, terminée par zéro (Contrôlez s'il s'agit vraiment de chiffres). Exemple: Entrée: 1 2 3 4 0 Affichage: 4321

**Exercice 10 :** Calculez le nombre lu à rebours d'un nombre positif entré au clavier en supposant que le fichier d'entrée standard contient le nombre à inverser. Exemple: Entrée: 1234 Affichage: 4321

**Exercice 11 :** Affichez un triangle isocèle formé d'étoiles de N lignes (N est fourni au clavier):

Nombre de lignes : 8

```
      *
     ***
    *****
   *********
  ***********
 *****
*****
*****
*****
```

**Exercice 4 :** affiche la table des produits pour N variant de 1 à 10 :

X*Y	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70

**Exercice 12 :** Calculer le pgcd de deux nombres a et b (pgcd : plus grand commun diviseur)

## Activité d'apprentissage 8

Module TRI-07 Techniques de programmation structurée

Précision C Optimiser l'algorithme

Code Activité C001

Activité Factoriser le code d'un programme en fonction et procédure

Durée 4 heures

Phase d'apprentissage Base

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Optimiser le code d'un programme en ayant recours aux fonctions et procédures.
- Utiliser la notion de récursivité dans un algorithme.

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 1: Écrire :

Une procédure, nommée f1, se contentant d'afficher bonjour (elle ne possèdera aucun argument) ;

Une procédure, nommée f2, qui affiche bonjour un nombre de fois égal à la valeur reçue en argument (entier);

Une fonction, nommée, f3, qui fait la même chose que f2, mais qui, de plus, renvoie la valeur (entier) 0.

**Exercice 2:** Écrire une fonction abs qui reçoit un entier x et retourne sa valeur absolue, écrire un algorithme qui exploite cette fonction.

**Exercice 3:** Écrire un programme se servant d'une fonction MOYENNE du type réel pour afficher la moyenne arithmétique de deux nombres réels entrés au clavier.

**Exercice 4:** Ecrire une fonction MIN et une fonction MAX qui déterminent le minimum et le maximum de deux nombres réels. Ecrire un programme se servant des fonctions MIN et MAX pour déterminer le minimum et le maximum de **quatre nombres réels entrés au clavier**.

**Exercice 5:** Écrire la fonction NCHIFFRES du type entier qui obtient une valeur entière N (positive ou négative) du type long entier comme paramètre et qui fournit le nombre de chiffres de N comme résultat.

Écrire un petit programme qui teste la fonction NCHIFFRES:

Exemple:

Introduire un nombre entier : 6457392

Le nombre 6457392 a 7 chiffres.

**Exercice 6:** En mathématiques, on définit la fonction factorielle de la manière suivante:

$0! = 1$

$n! = n*(n-1)*(n-2)* \dots * 1$  (pour  $n > 0$ )

Ecrire une fonction FACT du type **entier long** qui reçoit la valeur N (type **entier**) comme paramètre et qui fournit la factorielle de N comme résultat. Ecrire un petit programme qui teste la fonction FACT.

**Exercice 7 :** refaire l'exercice 6 en faisant appel à la récursivité.

**Exercice 8 :** Ecrire une fonction qui reçoit en arguments 2 nombres réels et un caractère et qui fournit un résultat correspondant) l'une des 4 opérations appliquées à ses deux premiers arguments, en fonction de la valeur de ce dernier, à savoir: addition pour le caractère +, soustraction pour le caractère -, multiplication pour \* et division pour / (tout autre caractère sera interprété comme une addition). On ne tiendra pas compte des risques de la division par 0.

## Activité d'apprentissage 9

Module	TRI-07	Techniques de programmation structurée
Précision	D	Utiliser des structures de données complexes.
Code Activité	D001	
Activité	• Les types complexes : les tableaux à une dimension	
Durée	8 heures	
Phase d'apprentissage	Base	

Détails sur les objectifs visés par l'activité	<p>Cette activité d'apprentissage doit vous permettre de :</p> <ul style="list-style-type: none"><li>▪ Manipuler des données complexes sous la forme de tableaux.</li><li>▪ Mettre au point des algorithmes complexes</li></ul>
--	---

Matière d'œuvre et/ou outillage	.....
	.....
	.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1:** Ecrire un algorithme qui remplit le tableau (de dimension 10) par des valeurs entrées au clavier et affiche le tableau. Calculer et afficher ensuite la somme des éléments du tableau.

**Exercice 2:** Écrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand et le plus petit parmi ces 10 nombres et leurs positions :

**Entrez le nombre numéro 1 : 12**

**Entrez le nombre numéro 2 : 14**

...

**Entrez le nombre numéro 10 : 6**

**Le plus grand de ces nombres est : 14, sa position : 2**

**Le plus petit de ces nombres est : 6, sa position 10**

**P.S : utilisez un tableau.**

**Exercice 3:** Soit T un tableau de 10 réels. Écrire l'algorithme qui permet de calculer le nombre des occurrences d'un nombre X (c'est-à-dire combien de fois ce nombre X figure dans le tableau T).

**Exercice 4:** Écrire un algorithme qui augmente de 1 tous les éléments d'un tableau, le nouveau tableau sera affiché à l'écran.

Exemple :  $\text{tab}[5] = \{1, 23, 6, 9, -1\}$  devient  $\text{tab}[5] = \{2, 24, 7, 10, 0\}$

**Exercice 5:** Écrire un programme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ.

**Exercice 6:** Écrire un programme qui permet de lire 10 notes et de déterminer le nombre de celles qui sont supérieures à la moyenne des notes. (En utilisant un tableau)

**Exercice 7:** Écrire un algorithme qui lit la dimension N d'un tableau T du type **entier** (dimension maximale: 20 éléments), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Effacer ensuite toutes les occurrences de la valeur 0 dans le tableau T et tasser les éléments restants.

**Exercice 8:** Écrire un algorithme qui lit la dimension N d'un tableau T du type **entier** (dimension maximale: 20 éléments), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Ranger ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide.

**Exercice 9:** Écrire un algorithme qui lit la dimension N d'un tableau T du type **Entier** (dimension maximale: 20 éléments), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau TPOS et toutes les valeurs strictement négatives dans un troisième tableau TNEG.

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 10:** Ecrire un programme qui demande à l'utilisateur de taper 10 entiers compris entre 0 et 20 qui seront stockés dans un tableau et qui affiche le nombre de fois qu'on a tapé un 0, le nombre de 1, le nombre de 2, ..., le nombre de 20.

**Exercice 11 :** On considère une séquence d'entiers  $s$  de longueur  $L$  représentée dans un tableau  $T$  d'entiers défini sur l'intervalle  $[1 \dots L_{\max}]$ ,  $0 < L < L_{\max}$ .

On veut écrire un programme qui remplace dans  $T$  la suite  $s$  par la suite  $s'$  de longueur  $L'$  (avec  $L' \leq L$ ), déduite de  $s$  en supprimant tous les éléments redondants. Un élément de  $s$  est redondant s'il est égal à un autre élément de  $s$ . Le programme ne doit pas utiliser de tableau intermédiaire pour créer  $s'$ . L'ordre des éléments reste celui de la séquence de départ. Etudier tout d'abord le problème en supposant que  $T$  peut ne pas être trié en ordre croissant.

Exemple : si  $s = [15, 4, 19, 4, 8, 11, 11, 3, 4, 19]$  et  $L = 10$  alors  $s' = [15, 4, 19, 8, 11, 3]$  et  $L = 6$

**Exercice 12:** Vérifier que deux tableaux sont identiques (les éléments des deux tableaux seront saisis par l'utilisateur).

**Exercice 13:** Rotation des éléments d'un tableau

Saisir un tableau  $tab[10]$  d'entiers;

Saisir le nombre de rotation

On effectue la rotation des éléments du tableau du nombre entré en faisant une rotation.

Afficher le tableau de départ, la valeur du déplacement et le vecteur à l'arrivée.

Exemple :

Départ 0 1 2 3 4 5 6 7 8 9

Rotations = 4

Arrivée 6 7 8 9 0 1 2 3 4 5

**Exercice 14 :**  $m$  personnes sont assises autour d'une table circulaire et entament le jeu suivant. Elles se fixent entre-elles un entier  $k > 0$  et une personne  $P_1$  par laquelle elles commenceront le jeu. On commence à compter de 1 à  $k$  en partant de  $P_1$  et en tournant à droite. La  $k^{\text{ième}}$  personne quitte la table et on continue à compter de 1 à  $k$  à partir de la personne suivante encore à table. La personne gagnante est la dernière qui reste à table.

Analyser ce problème et faire un programme permettant de réaliser ce jeu.



## Activité d'apprentissage 10

Module TRI-07 Techniques de programmation structurée

Précision D Utiliser des structures de données complexes.

Code Activité D002

Activité • Les types complexes : les tableaux à deux dimensions

Durée 4 heures

Phase d'apprentissage Base

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Utiliser des structures de données complexes.
- Travailler avec des tableaux à plusieurs dimensions

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui lit les dimensions L et C d'un tableau T à deux dimensions du type entier (dimensions maximales: 10 lignes et 10 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

Exercice

**Exercice 2:** Les points cols d'un tableau à deux dimensions sont les éléments du tableau qui sont minimum sur leur ligne et maximum sur leur colonne.

Ecrire le programme principal qui recherche dans un tableau à deux dimensions les points cols.

Exemple : soit le tableau suivant  $T[3][3]$

```
1  2  3
-7  8  9
-6 -3 -4
```

Les points cols du tableau sont :

$T[0][0]=1$

$T[2][1]=-3$

**Exercice 3:** Un carré magique est un carré rempli de nombres qui, lorsque l'on en fait la somme sur chaque ligne, colonne ou diagonale, donne le même résultat. Pour simplifier le travail nous ne considérerions ici que des carrés d'ordre impair.

Écrire un algorithme qui teste si un carré est magique.

Exemple de carré d'ordre 3 :

```
8 1 6
3 5 7
4 9 2
```

**Exercice 4:**

1. Lire une matrice de N lignes et M colonnes ligne par ligne. Les données sont dans l'ordre :

```
N  M
A[1][1] ... A[1][M]
...
A[N][1] ... A[N][M]
```

Écrire ensuite cette matrice ligne par ligne.

Les exercices suivants supposent que cette phase d'initialisation vient d'être faite.

2. Compter (et afficher) le nombre de zéros de chaque colonne.

3. Compter (et afficher) le nombre de colonnes contenant au moins un zéro.

4. Afficher le minimum de chaque colonne.

5. Afficher le numéro de la colonne contenant le minimum de la matrice. (A égalité, donner le plus petit numéro de colonne).

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 5 :

1. Écrire un programme qui transfère une matrice M à deux dimensions L et C (dimensions maximales: 10 lignes et 10 colonnes) dans un tableau V à une dimension L\*C.

**Exemple:**

$$\begin{pmatrix} / & & \backslash \\ | & a & b & c & d & | \\ | & e & f & g & h & | \\ | & i & j & k & l & | \\ \backslash & & / \end{pmatrix} \Rightarrow \begin{pmatrix} / & & \backslash \\ | & a & b & c & d & e & f & g & h & i & j & k & l & | \\ \backslash & & / \end{pmatrix}$$

2. Écrire un programme qui effectue la transposition  ${}^tA$  d'une matrice A de dimensions N et M en une matrice de dimensions M et N.

**Rappel:**

$${}^tA = {}^t \begin{pmatrix} / & & \backslash \\ | & a & b & c & d & | \\ | & e & f & g & h & | \\ | & i & j & k & l & | \\ \backslash & & / \end{pmatrix} = \begin{pmatrix} / & & \backslash \\ | & a & e & i & | \\ | & b & f & j & | \\ | & c & g & k & | \\ | & d & h & l & | \\ \backslash & & / \end{pmatrix}$$

3. Écrire un programme qui réalise l'addition de deux matrices A et B de mêmes dimensions N et M.

**Rappel:**

$$\begin{pmatrix} / & & \backslash \\ | & a & b & c & d & | \\ | & e & f & g & h & | \\ | & i & j & k & l & | \\ \backslash & & / \end{pmatrix} + \begin{pmatrix} / & & \backslash \\ | & a' & b' & c' & d' & | \\ | & e' & f' & g' & h' & | \\ | & i' & j' & k' & l' & | \\ \backslash & & / \end{pmatrix} = \begin{pmatrix} / & & \backslash \\ | & a+a' & b+b' & c+c' & d+d' & | \\ | & e+e' & f+f' & g+g' & h+h' & | \\ | & i+i' & j+j' & k+k' & l+l' & | \\ \backslash & & / \end{pmatrix}$$

**Exercice 6 :** Écrire un programme qui effectue la multiplication de deux matrices A et B. Le résultat de la multiplication sera mémorisé dans une troisième matrice C qui sera ensuite affichée.

## Activité d'apprentissage 11

Module	TRI-07	Techniques de programmation structurée
Précision	D	Utiliser des structures de données complexes.
Code Activité	D003	
Activité	• Les types complexes : les chaînes de caractères	
Durée	3 heures	
Phase d'apprentissage	Base	

Détails sur les objectifs visés par l'activité	<p>Cette activité d'apprentissage doit vous permettre de :</p> <ul style="list-style-type: none"><li>▪ Manipuler des données complexes sous la forme de tableaux.</li><li>▪ Mettre au point des algorithmes complexes.</li><li>▪ Travailler avec des tableaux de caractères.</li></ul>
--	--

Matière d'œuvre et/ou outillage	.....
	.....
	.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Écrire un algorithme qui lit une phrase, la mémorise dans une variable chaîne et affiche ensuite:

- a) la longueur L de la chaîne.
- b) le nombre de 'a' contenus dans le texte.
- c) toute la phrase à rebours, sans changer le contenu de la variable chaîne.
- d) toute la phrase à rebours, après avoir inversé l'ordre des caractères dans chaîne.

**Exercice 2:** Écrire un algorithme qui supprime les espaces et tabulations d'une chaîne de caractères. (sans utiliser de tableau intermédiaire).

**Exercice 3:** Écrire un algorithme qui permet de dire si deux chaînes de caractères sont identiques ou pas.

**Exercice 4:** Écrire un algorithme qui permet de concaténer deux chaînes de caractères.

**Exercice 5:** Vous devez déterminer si une chaîne de caractères est un palindrome.

Un palindrome est une chaîne de caractères que l'on peut lire identiquement de droite à gauche, et de gauche à droite.

Par exemple :

AA.

38783.

LAVAL.

LAVAL A ETE A LAVAL.

**Exercice 6 :** vous devez compter le nombre d'occurrence de 2 lettres consécutives dans la chaîne. Un couple trouvé dans la phrase est éliminé et ne sera comptabilisé qu'une seule fois, comme dans l'exemple suivant :

Phrase = « fonctions creees par le programmeur.

Couple cherché = « ee »

Résultat = 1

**Exercice 7 :** Écrire un programme qui lit 5 mots, séparés par des espaces et qui les affiche ensuite dans une ligne, mais dans l'ordre inverse. Les mots sont mémorisés dans un tableau de chaînes de caractères.

Exemple

voici une petite phrase !

! phrase petite une voici



**Exercice 8 :** Écrire un algorithme qui lit un verbe régulier en "er" au clavier et qui en affiche la conjugaison au présent de l'indicatif de ce verbe. Contrôlez s'il s'agit bien d'un verbe en "er" avant de le conjugue.

Exemple:

Verbe : chanter  
je chante  
tu chantes  
il chante  
nous chantons  
vous chantez  
ils chantent

**Exercice 9 :** Un des plus anciens systèmes de cryptographie (aisément déchiffrable) consiste à décaler les lettres d'un message pour le rendre illisible. Ainsi, les A deviennent des B, les B des C, etc. Écrire un algorithme qui demande une phrase à l'utilisateur et qui la code selon ce principe.

**Exercice 10 :** Écrire un programme qui demande à l'utilisateur de taper une phrase et un mot, puis la programme affiche si le mot été retrouvé dans la phrase ou pas :

Exemple :

phrase : « les stagiaires de la TRIIE sont tous intelligents »

Mot recherché « TRIIE »

Le mot existe dans la phrase.

## Activité d'apprentissage 12

Module TRI-07 Techniques de programmation structurée

Précision D Utiliser des structures de données complexes.

Code Activité D004

Activité • Les types complexes : les enregistrements et les tableaux d'enregistrements

Durée 3 heures

Phase d'apprentissage ENTRAINEMENT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Manipuler des données complexes sous la forme de tableaux.
- Mettre au point des algorithmes complexes
- Travailler avec des enregistrements et des tableaux d'enregistrements.

Matière d'œuvre  
et/ou outillage

.....

.....

.....

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Dans une classe un élève est connu par :

- Son nom
- Son prénom
- Son numéro d'inscription
- Sa date de naissance
- Sa note d'examen

On désire un algorithme permettant de faire les opérations suivantes :

- Saisie au clavier des informations de NE élèves
- Calcul de :
  - ✓ La somme totale des notes de toute la classe STN.
  - ✓ La moyenne de la classe MOY.
  - ✓ Le pourcentage de la note de chaque élève dans la somme totale des notes PN.
  - ✓ Tri du tableau des élèves par ordre croissant des notes.
- Affichage à l'écran
  - ✓ Des données de chaque élève
  - ✓ De tous les résultats calculés par l'algorithme.

**Exercice 2:**

Demain est un autre jour. Nous supposons que l'utilisateur ne rentre que des entiers strictement positifs.

1) Définir le type date basé sur une structure permettant de représenter une date comportant le numéro du jour, le numéro du mois et l'année.

Ex : 14 7 1792 pour le 14 juillet 1792

25 12 2007 pour le 25 décembre 2007

29 2 2008 pour le 29 février 2008

2) Écrire la fonction `estBissextile()` qui renvoie 1 si l'année est bissextile; 0 sinon. Nous rappelons qu'une année est bissextile si elle est divisible par 4 mais pas par 100, elle est aussi bissextile si elle est divisible par 400.

3) Écrire la fonction `dateJuste()` qui renvoie 1 si une date saisie est juste, 0 sinon

4) Écrire la procédure `jourdulendemain()` qui à partir d'une donnée de type date détermine la date du lendemain.

5) Écrire une fonction `comparedate()` qui renvoie 1 si une date d1 est plus grande qu'une date d2, 0 si elles sont égales, et -1 si la date d2 est plus grande que la date d1

6) Écrire une fonction `ecartjour()` qui retourne le nombre de jours qui séparent deux dates.



## Activité d'apprentissage 13

Module	TRI-07	Techniques de programmation structurée
Précision	D	Utiliser des structures de données complexes.
Code Activité	D005	
Activité	Les algorithmes de tris.	
Durée	8 heures	
Phase d'apprentissage	ENTRAINEMENT	

Détails sur les objectifs visés par l'activité	<p>Cette activité d'apprentissage doit vous permettre de :</p> <ul style="list-style-type: none"><li>▪ Manipuler des données complexes sous la forme de tableaux.</li><li>▪ Mettre au point des algorithmes complexes</li><li>▪ Mettre en œuvre les algorithmes de tris.</li></ul>
--	--

Matière d'œuvre et/ou outillage	.....
	.....
	.....

## DESCRIPTION DE L'ACTIVITÉ



### 1- Tri par sélection

Vous devez trier un tableau de 10 entiers dont les valeurs vous sont fournies. On vous demande de respecter scrupuleusement le processus de tri suivant, qualifié de tri par sélection. Vous devez chercher le plus petit élément du tableau et le placer en 1er, puis rechercher le plus petit parmi les valeurs non triées et le placer en second, etc...

Valeurs initiales sont chargées en début de programme :

52 10 1 25 62 3 8 55 3 23

Vous ne devez utiliser que ce seul tableau.

Exemples de résultats de la progression :

1 52 10 25 62 3 8 55 3 23

1 3 52 10 25 62 8 55 3 23

1 3 3 52 10 25 62 8 55 23

1 3 3 8 52 10 25 62 55 23

1 3 3 8 10 52 25 62 55 23

1 3 3 8 10 23 52 25 62 55

1 3 3 8 10 23 25 52 62 55

1 3 3 8 10 23 25 52 62 55

1 3 3 8 10 23 25 52 55 62

### 2- Tri Bulle :

On vous demande cette fois de trier les éléments selon le principe du tri bulle.

Le tri bulle est un tri plus astucieux et consiste à faire remonter petit à petit un élément trop petit vers le haut du tableau en comparant les éléments deux à deux.

Si l'élément de gauche est supérieur à son voisin de droite, ils sont inversés et le programme poursuit avec le suivant. Lorsque le programme atteint la fin du tableau, le traitement reprend du début. Le programme s'arrête quand tous les éléments sont bien placés.

Liste initiale :

52 10 1 25 62 3 8 55 3 23

Exemples de résultats de la progression :

10 52 1 25 62 3 8 55 3 23

10 1 52 25 62 3 8 55 3 23

10 1 25 52 62 3 8 55 3 23

10 1 25 52 62 3 8 55 3 23

10 1 25 52 3 62 8 55 3 23

10 1 25 52 3 8 62 55 3 23

10 1 25 52 3 8 55 62 3 23

10 1 25 52 3 8 55 3 62 23

10 1 25 52 3 8 55 3 23 62

L'ensemble du tableau a été parcouru. Le programme recommence jusqu'à ce que tous les éléments soient bien triés. Là encore, vous ne devez travailler qu'avec ce seul tableau.



### 3- Tri par permutation :

Vous devez maintenant effectuer ce tri par permutation des valeurs.

Le tri par permutation est similaire à celui du tri d'un jeu de cartes.

Le tableau jusqu'à ce qu'un élément plus petit que le précédent, donc mal placé, soit trouvé. Cet élément est alors rangé à sa place dans le tableau et les autres valeurs déplacées, puis la lecture des autres éléments reprend. Le programme s'arrête à la fin du tableau.

Liste initiale chargée initialement :

52 10 1 25 62 3 8 55 3 23

Exemples de progressions

10 52 1 25 62 3 8 55 3 23

1 10 52 25 62 3 8 55 3 23

1 10 25 52 62 3 8 55 3 23

1 3 10 25 52 62 8 55 3 23

1 3 8 10 25 52 62 55 3 23

1 3 8 10 25 52 55 62 3 23

1 3 3 8 10 25 52 55 62 23

1 3 3 8 10 23 25 52 55 62

### 4- Tri par comptage :

Le tri par comptage consiste pour chaque élément du tableau à compter combien d'éléments sont plus petits que lui, grâce à ce chiffre on connaît sa position dans le tableau résultat.

Prenez garde aux valeurs identiques ! Pour réaliser ce tri, utilisez deux tableaux

### 5- Tri Fusion :

On dispose de deux tableaux A et B (de dimensions respectives N et M), triés par ordre croissant. Fusionner les éléments de A et B dans un troisième tableau FUS trié par ordre croissant.

## Activité d'apprentissage 14

Module TRI-07 Techniques de programmation structurée

Précision D Utiliser des structures de données complexes.

Code Activité D006

Activité Les algorithmes de recherche rapide.

Durée 4 heures

Phase d'apprentissage ENTRAINEMENT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Manipuler des données complexes sous la forme de tableaux.
- Mettre au point des algorithmes complexes
- Mettre en œuvre un algorithme de recherche rapide (et récursive)

Matière d'œuvre et/ou outillage

.....

.....

.....

## DESCRIPTION DE L'ACTIVITÉ



### Recherche séquentielle :

On suppose qu'on dispose d'un tableau trié par ordre croissant de dimension  $N$ .  
Ecrire un algorithme qui permet de rechercher une valeur  $X$  dans un tableau. Si la valeur recherché est trouvé, la recherche doit prendre fin.

Idée : on peut arrêter de chercher avant la fin (si l'on trouve une valeur supérieur à la valeur recherché)

### Recherche dichotomique :

Il existe plusieurs méthodes pour faire une recherche d'une variable dans un tableau. L'une des plus importantes est appelée recherche dichotomique. Ce type de recherche est utilisé que si le tableau est trié :

2	7	19	84	190
---	---	----	----	-----

Le principe est de comparer la valeur recherchée  $X$  par celle du milieu du tableau  $T[N/2]$  ;

Si  $T[N/2]=X$  on arrête car on a trouvé ce qu'on cherche, c'est  $T[N/2]$  ;

Si  $X > T[N/2]$ , refaire la recherche sur la 2<sup>ème</sup> partie du tableau

Si  $X < T[N/2]$ , refaire la recherche sur 1<sup>ère</sup> partie du tableau

Si la dimension du tableau est paire, on prend l'élément équivalent à la valeur absolue majoritaire.

Écrire l'algorithme qui permet d'effectuer la recherche dichotomique dans un tableau.

## Activité d'apprentissage 15

Module	TRI-07	Techniques de programmation structurée
Précision	D	Utiliser des structures de données complexes.
Code Activité	D007	
Activité	Les structures de données complexes	
Durée	8 heures	
Phase d'apprentissage	TRANSFERT	

Détails sur les objectifs visés par l'activité	<p>Cette activité d'apprentissage doit vous permettre de :</p> <ul style="list-style-type: none"><li>▪ Manipuler des données complexes sous la forme de tableaux.</li></ul>
--	---

Matière d'œuvre et/ou outillage	.....
	.....
	.....

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 1 :

Le directeur d'une société offre des primes à ses employés. Chaque employé possède les informations suivantes :

CIN, Nom, Prénom, Date de naissance, Date d'embauche, Salaire, Nombre d'année d'expérience (NAE), prime.

Les primes sont offertes selon le critère suivant :

Si nombre d'année d'expérience  $2 < \text{NAE} \leq 4$  alors  $\text{prime} = \text{salaire}$

Si nombre d'année d'expérience  $4 < \text{NAE} \leq 6$  alors  $\text{prime} = \text{salaire} * 1,5$

Si nombre d'année d'expérience  $6 < \text{NAE} \leq 8$  alors  $\text{prime} = \text{salaire} * 2$

Si nombre d'année d'expérience  $\text{NAE} > 8$  alors  $\text{prime} = \text{salaire} * 4$

### Questions

- 1- Créer les structures nécessaires avec les attributs cités dans l'énoncé, en choisissant un type de données convenable pour chaque attribut.
- 2- Remplir toutes les informations des 6 employés sauf le NAE et la prime.
- 3- Calculer le nombre d'année d'expérience de chaque employé.
- 4- Ecrire une fonction ayant comme paramètre le salaire, retourne la prime. Utiliser cette fonction pour remplir la valeur de prime de chaque employé.
- 5- Afficher tous les employés qui ont été embauché avant la date 01/04/2005 et âgés d'au moins de 30 ans.
- 6- Trier la liste des employés selon leur salaire.

### Exercice 2 :

Une société commercialise 8 produits dans 3 magasins différents.

On désire faire les statistiques de ventes annuelles. Pour cela, on utilise trois tableaux : V1, V2 et V3 représentant respectivement la quantité vendue dans l'année du  $i^{\text{ème}}$  produit dans les magasins 1, 2 et 3.

L'étude consiste à établir les programmes suivants :

1) Procédure de saisie au clavier de toutes les données (remplissage des tableaux de quantités vendues V1, V2 et V3.

2) Procédure de calcul de la quantité totale vendue par produit, tout magasin réunis.

Stocker ces résultats dans un tableau M.

3) Procédure de calcul de la quantité totale vendue par magasin, tout produit réunis.

Stocker ces résultats dans un tableau P.

4) Procédure de calcul pour chaque produit la moyenne vendue. Stocker ces résultats dans un tableau Y.



### Exercice 3 :

Le but est de créer un ensemble de fonction de gestion des nombres complexes. On rappelle que les nombre complexes d'écrivent comme suit :  $x=a+ib$ , où  $a$  et  $b$  sont des nombres entiers,  $i*i=-1$ .

- 1- Définir la structure de données nécessaires pour manipuler les nombres complexes.
- 2- Ecrire une fonction permettant de prendre deux nombres complexes et de retourner un nombre complexe représentant la somme. Rappel si  $x=a+ib$  et  $y=c+id$  alors  $z=x+y=a+c+i(b+d)$
- 3- Ecrire une fonction permettant de prendre deux nombres complexes et de retourner un nombre complexe représentant le produit. Rappel si  $x=a+ib$  et  $y=c+id$  alors  $z=x*y=(ac-bd)+i(ad+bd)$
- 4- Ecrire une procédure permettant d'afficher un nombre complexe.

### Exercice 4 :

Chaque élève d'une classe s'est vu attribuer dix notes correspondant à dix matières (une note fictive égale à -1 correspond à une absence lors de l'épreuve de contrôle). A chacune de ces matières correspond un coefficient.

On suppose que le nombre d'élèves ne dépasse pas 50.

- Pour tous les calculs de moyenne, on ne tient pas compte des absences.
- On ne calcule la moyenne effective d'un élève que s'il a été présent au moins à un contrôle.
- On note par -1 la moyenne d'un élève absent à tous les contrôles.

Nous allons utiliser un tableau NOTE de 50 lignes et 10 colonnes. Chaque ligne correspond à un élève, et chaque colonne à une matière. Le tableau MOY de 50 lignes permet de stocker la moyenne de chaque élève. On utilise un tableau COEF de 10 lignes, dans lequel on stocke le coefficient de chaque matière.

L'ordre dans COEF correspond à l'ordre des colonnes de NOTE: par exemple si la première colonne de NOTE correspond aux notes de mathématique, on doit trouver en COEF[1] le coefficient mathématique.

0. Déclarer les tableaux et remplissez-les.

Écrire un algorithme qui permettrait de calculer:

1. La moyenne de chaque élève.
2. La moyenne générale de la classe
3. Pour chaque matière, la moyenne de la classe et le pourcentage d'absentéisme.



## Activité d'apprentissage 16

Module TRI-07 Techniques de programmation structurée

Précision E Manipuler les fichiers de données.

Code Activité E001

Activité Déclaration des fichiers

Durée 4 heures

Phase d'apprentissage BASE

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Manipuler des fichiers texte à accès séquentiel.

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

---

## DESCRIPTION DE L'ACTIVITÉ



**Exercice 1 :** Créer puis afficher à l'écran le fichier INFORM.TXT dont les informations sont structurées de la manière suivante:

Numéro de matricule (entier)

Nom (chaîne de caractères)

Prénom (chaîne de caractères)

Le nombre d'enregistrements à créer est à entrer au clavier par l'utilisateur.

**Exercice 2 :** Ecrire un programme qui crée un fichier INFBIS.TXT qui est la copie exacte (enregistrement par enregistrement) du fichier INFORM.TXT.

**Exercice 3 :** Ajouter un nouvel enregistrement (entré au clavier) à la fin de INFORM.TXT et sauver le nouveau fichier sous le nom INFBIS.TXT.

**Exercice 4 :** Supprimer dans INFORM.TXT tous les enregistrements:

a) dont le numéro de matricule se termine par 8

b) dont le prénom est "Leila"

c) dont le nom est un palindrome. Définir une fonction d'aide PALI qui fournit le résultat 1 si la chaîne transmise comme paramètre est un palindrome, sinon la valeur zéro.

Sauver le nouveau fichier à chaque fois sous le nom INFBIS.TXT.

**Exercice 5 :**

A l'aide d'un éditeur de textes, créer un fichier NOMBRES.TXT qui contient une liste de nombres entiers. Dans le fichier, chaque nombre doit être suivi par un retour à la ligne. Ecrire un programme qui affiche les nombres du fichier, leur somme et leur moyenne.

## Activité d'apprentissage 17

Module TRI-07 Techniques de programmation structurée

Précision E Manipuler les fichiers de données.

Code Activité E002

Activité Manipulation des fichiers

Durée 2 heures

Phase d'apprentissage ENTRAINEMENT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Manipuler des fichiers à accès séquentiel.

Matière d'œuvre et/ou outillage

.....

.....

.....

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 1 :

Vous allez devoir écrire un programme qui permet de créer, mettre à jour ou supprimer un enregistrement client.

Le nom du fichier ainsi que son emplacement sur disque sont fournis par l'utilisateur du programme. Si le fichier n'existe pas, vous devez le créer. Pensez à optimiser votre code en ayant recours aux fonctions et procédures de manière adéquate.

La structure d'un enregistrement d'un fichier client est constitué de :

- Le numéro client : ENTIER qui correspond au N° d'ordre d'enregistrement dans le fichier
- Le caractère de validité de l'enregistrement : de type CAR. Il vaut 'X' si l'enregistrement a été supprimé. Ces enregistrements ne devront alors pas être considérés pour lecture ou modification.
- Le prénom du client : de type CHAÎNE de longueur 30
- Le nom du client : de type CHAÎNE de longueur 30
- L'adresse du client Partie 1 CHAÎNE de longueur 40
- L'adresse du client Partie 2 CHAÎNE de longueur 40
- Le code postal de la ville du client : CHAÎNE de longueur 5

La ville du client : CHAÎNE de longueur 34

Optimiser le code du programme en ayant recours aux fonctions et procédures.

L'utilisateur doit indiquer s'il doit insérer un nouvel enregistrement, mettre à jour ou supprimer un enregistrement existant. Dans le cas d'une suppression, vous devez afficher les informations du client et demandez la confirmation de cette suppression.

Pour la modification ou la suppression, vous devez accéder à l'enregistrement par le N° d'ordre qui doit être compris entre 1 et le nombre d'enregistrements stockés dans le fichier.

Pour l'ajout, vous devez attribuer un nouveau n° ordre en incrémentant le nombre d'enregistrements stockés dans le fichier.

Pour une mise à jour, vous devez afficher l'ancienne valeur devant être modifiée.

## Activité d'apprentissage 18

Module TRI-07 Techniques de programmation structurée

Précision E Manipuler les fichiers de données.

Code Activité E003

Activité Manipulation des fichiers

Durée 9 heures

Phase d'apprentissage TRANSFERT

Détails sur les objectifs visés par l'activité

Cette activité d'apprentissage doit vous permettre de :

- Manipuler des fichiers à accès séquentiel.

Matière d'œuvre  
et/ou outillage

.....  
.....  
.....

## DESCRIPTION DE L'ACTIVITÉ



### Exercice 1 :

Soit un fichier texte contenant un ensemble de mots séparés par des retours à la ligne (que l'on va qualifier de dictionnaire)

Ecrire les fonctions et les procédures suivantes :

- 1- Charger(chaine mot), qui stocke dans la chaîne un mot du fichier choisi au **hasard** par l'ordinateur.

Nb : on suppose l'existence d'une fonction hasard() qui génère un nombre aléatoire entier.

- 2- Code(chaine mot, chaine res) qui code une chaîne de caractère conformément à l'exemple : BONJOUR Devint B-----R

- 3- Rechercher(chaine mot, chaine mot\_code, caractère lettre) qui permet à partir d'un mot, d'une chaîne codée et d'une lettre donnée, de placer dans la chaîne codée la lettre à toutes les positions où elle se trouve dans le mot, si la lettre n'est pas dans le mot on retournera un état non trouvé.

- 4- JEU() qui permet à un joueur de proposer une lettre, et de compter le nombre des propositions faites.

- ✓ On cherchera alors dans une chaîne chargée la présence de la lettre.
- ✓ Si la lettre est présente on la placera dans une chaîne codée à toutes les positions où elle se situe dans la chaîne saisie.
- ✓ Si la lettre n'est pas présente on comptera une erreur.
- ✓ Le jeu se finira sur le fait que toutes les lettres de la chaîne aient été trouvées ou que le nombre d'erreurs soit égal à 8.



## Exercice 2 :

Outil de Traitement de texte

Il s'agit de mettre en place une application pour le traitement de texte

Le projet doit contenir le menu suivant :

Tout d'abord saisir le nom du fichier texte au clavier puis accéder au menu suivant :

- 1- Analyse du texte
- 2- Consultation du texte
- 3- Recherche d'un mot
- 4- Remplacement d'un mot
- 5- Suppression d'un mot
- 6- Sauvegarder
- 7- Quitter

Pour le menu Analyse du texte : le programme doit afficher le nombre de caractères qu'il contient,

- le nombre de chacune des lettres de l'alphabet (sans distinguer les majuscules et les minuscules),
- le nombre de mots,
- le nombre de paragraphes (c.-à-d.: des retours à la ligne),

Les retours à la ligne ne devront pas être comptabilisés dans les caractères. On admettra que deux mots sont toujours séparés par un ou plusieurs des caractères suivants:

- fin de ligne
- espace
- ponctuation: . : , ; ? !
- parenthèses: ( )
- guillemets: "
- apostrophe: '

Utiliser une fonction d'aide SEPA qui décide si un caractère transmis comme paramètre est l'un des séparateurs mentionnés ci-dessus. SEPA restituera la valeur (logique) 1 si le caractère est un séparateur et 0 dans le cas contraire. SEPA utilise un tableau qui contient les séparateurs à détecter.

Exemple:

```
Nom du fichier texte : A:LITTERA.TXT
Votre fichier contient:
  12 paragraphes
  571 mots
  4186 caractères
dont
  279 fois la lettre a
  56 fois la lettre b
  . . .
  3 fois la lettre z
et 470 autres caractères
```



### Exercice 3 :

#### Gestion de personnel

Il faut créer un fichier où vous avez toutes les informations nécessaires concernant des personnes.

Une personne est connue par son : nom, prénom, sexe, date de naissance (jour, mois, année), un identifiant (5 caractères ex : P0001), le nombre d'enfants et leurs identifiants.

Les traitements (fonctions) qu'il faut faire :

- ✓ Charger les données du fichier dans une liste des personnes (en utilisant une structure personne).
- ✓ Ajouter une personne dans la liste (donnée en paramètre).
- ✓ Ajouter les enfants d'une personne (donnée en paramètre).
- ✓ Supprimer une personne de la liste.
- ✓ Consulter la liste des personnes.
- ✓ Chercher une personne par son nom ou prénom.
- ✓ Afficher les informations sur les enfants d'une personne donnée en paramètre.
- ✓ Retourner le nom de mère ou père d'une personne. (le nom de la personne est donné en paramètre)
- ✓ Retourner l'âge d'une personne.
- ✓ Modifier les enregistrements d'une personne. (Mise à jour).
- ✓ Trier la liste des personnes (selon le nom et l'identifiant).
- ✓ Sauvegarder les données de la liste dans un fichier
- ✓ Toutes autres fonctions que vous considérez utiles ...

Il faut toujours donner le choix à l'utilisateur. Ex à titre d'exemple : création d'un menu