

Révissez Votre Bac

Notre site « www.BAC.org.tn » vous donne accès à :

1- Des Examens de baccalauréat

2- Des Devoirs de contrôle et synthèse " Sfax et Autres "

3- Des Cours et des résumés " Facile A comprendre "

4- Des Séries avec corrigés

5- Des Quiz et des tests d'intelligence avec score

6- Des Groupes de discussion privée pour résoudre vos problèmes

7- Vous Pouvez Gagnés D'argent Facilement



EXERCICES

Exercice N°1 :

Utiliser les mots suivants pour compléter le texte :

réseaux, données, informations, mémoire de masse, fichiers, DataBase

Une base de données (.....) est un ensemble structuré et organisé de permettant le stockage de grandes quantités d'..... afin d'en faciliter l'exploitation.

Une base de données se traduit physiquement par un ensemble de présent sur une (bien souvent un disque). Certaines peuvent être accessibles via les, on parle alors de base de données en ligne.

Exercice N°2 :

Mettre dans la case correspondante la lettre (V) si la réponse est correcte et (F) sinon.

Lorsque j'utilise une base de données, je manipule (je créé, je modifie...)

<input type="checkbox"/>	Des tables
<input type="checkbox"/>	Des cellules
<input type="checkbox"/>	Des feuilles de calcul

Une base de données garde les informations d'une façon :

<input type="checkbox"/>	Volatile
<input type="checkbox"/>	Temporaire
<input type="checkbox"/>	Permanente

Une base de données est :

<input type="checkbox"/>	Un ensemble de données appartenant au même sujet
<input type="checkbox"/>	Un serveur Internet
<input type="checkbox"/>	Sauvegardée sur une mémoire de masse (disque dur)

On peut Gérer une base de données à travers :

<input type="checkbox"/>	Un seul ordinateur
<input type="checkbox"/>	Une architecture client/serveur
<input type="checkbox"/>	Un serveur Web

Exercice N°3 :

- Proposer deux inconvénients liés à l'utilisation des fichiers.
- Citer deux intérêts de l'utilisation des bases de données.
- Citer quatre fonctionnalités d'un SGBD.

Exercice N°4 :

Pour chacune des propositions suivantes, écrivez (V) si la déclaration est permise, sinon écrivez (F) et justifiez brièvement votre réponse.

- Un SGBD permet un accès facile et convivial aux données.
- Un LMD permet la description des données de la base.
- La gestion des accès concurrent consiste à verrouiller l'accès aux données lors d'une consultation.

Exercice N°5 :**Mettre une croix devant la ou les bonnes réponses :**

1) Un SGBD relationnel est basé sur :

- Le concept de classes
 Le concept de tables
 Le moteur de recherche

2) Un domaine correspond à :

- Une liste de valeurs de différents types
 L'ensemble des valeurs d'un attribut
 L'ensemble des propriétés des colonnes

3) Une BD garde les Informations de façon:

- Volatile
 Permanente
 Temporaire

4) Une contrainte d'intégrité est :

- Une condition d'arrêt
 Une condition qui doit être toujours vérifiée
 Appliquée à une colonne, table ou même à un domaine

Exercice N°6 :**Compléter les cases du tableau suivant :**

Terme	Définition
.....	Propriété fondamentale de la base permettant la cohérence des données.
.....	Basé sur les liens père-fils (1 : n).
.....	Machine qui gère une base de données en architecture clients/serveur.
.....	Basé sur la représentation des objets et des liens à l'aide de tables.
.....	Partage des données par différents utilisateurs en même temps.
.....	Structure permettant d'éviter la redondance en saisissant une seule fois les données d'une base.

Exercice N°7 :**Mettre dans la case correspondante la lettre (V) si la réponse est correcte et (F) sinon.****Parmi les fonctions d'un SGBD :**

- Sécurisation des données
 Concurrence d'accès aux données
 La gestion des ressources matérielles et logicielles

Parmi les SGBD, on a :

- Microsoft Word
 ACCESS
 MySQL
 Macromédia Flash

La sécurité en base de données signifie :

- Sauvegarde périodique de la base de données
- La définition des données
- Restauration de la base en cas de panne
- Manipulation des données

Exercice N°8 :

Compléter la grille suivante de manière horizontale par des mots correspondants aux définitions suivantes :

A	B	C	<u>D</u>	E	F	G	H	I	J	K	L
---	---	---	----------	---	---	---	---	---	---	---	---

1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											

- 1) Champ ou attribut.
- 3) Stocker des données
- 5) Capacité de stocker et de retrouver une donnée
- 7) Gérer une Base de Données
- 9) Composante d'un SGBD
- 11) Exemple de SGBD permettant de créer et de gérer des BD

NB : les autres lettres sont à déduire.

Définir le mot trouvé de la colonne (**D**) :

.....

.....

.....

CORRECTION

Exercice N° 1 :

Une base de données (**DataBase**) est un ensemble structuré et organisé de **données** permettant le stockage de grandes quantités d'**informations** afin d'en faciliter l'exploitation.

Une base de données se traduit physiquement par un ensemble de **fichiers** présents sur une **mémoire de masse** (bien souvent un disque). Certaines peuvent être accessibles via les **réseaux**, on parle alors de base de données en ligne.

Exercice N°2 :

V
F
F

F
F
V

V
F
V

V
V
V

Exercice N°3 :

- a) Proposer deux inconvénients liés à l'utilisation des fichiers :
 - Lourdeurs d'accès
 - Redondance de données
- b) Citer deux intérêts de l'utilisation des bases de données :
 - Centralisation des données
 - Partage des données
- c) Citer quatre fonctionnalités d'un SGBD:
 - Définition des données
 - Manipulation des données
 - Gestion des accès
 - Sécurité

Exercice N°4 :

- a) V
- b) F → La description des données est assurée par le LDD "Langage de description des données" par contre la manipulation est assurée par Le LMD, elle consiste à interroger les données de la base à travers des requêtes.

- c) F → Lors d'une consultation d'une donnée un SGBD doit autoriser l'accès simultané par contre lors d'une modification il doit assurer le verrouillage des données.

Exercice N°5 :

Mettre une croix devant la ou les bonnes réponses :

1) Un SGBD relationnel est basé sur :

2) Un domaine correspond à :

<input type="checkbox"/>	Le concept de classes
<input checked="" type="checkbox"/>	Le concept de tables
<input type="checkbox"/>	Le moteur de recherche

<input type="checkbox"/>	Une liste de valeurs de différents types
<input checked="" type="checkbox"/>	L'ensemble des valeurs d'un attribut
<input checked="" type="checkbox"/>	L'ensemble des propriétés des colonnes

3) Une BD garde les informations de façon:

4) Une contrainte d'intégrité est :

<input type="checkbox"/>	Volatile
<input checked="" type="checkbox"/>	Permanente
<input type="checkbox"/>	Temporaire

<input type="checkbox"/>	Une condition d'arrêt
<input checked="" type="checkbox"/>	Une condition qui doit être toujours vérifiée
<input checked="" type="checkbox"/>	Appliquée à une colonne, table ou même à un domaine

Exercice N°6 :

Compléter les cases du tableau suivant :

Terme	Définition
Intégrité des données	Propriété fondamentale de la base permettant la cohérence des données.
Modèle hiérarchique	Basé sur les liens père-fils (1 : n).
Serveur de données	Machine qui gère une base de données en architecture clients/serveur.
Modèle relationnel	Basé sur la représentation des objets et des liens à l'aide de tables.
Concurrence d'accès	Partage des données par différents utilisateurs en même temps.
Centralisation	Structure permettant d'éviter la redondance en saisissant une seule fois les données d'une base.

Exercice N°7 :

V
V
F

F
V

V
F

V
F
V
F

Exercice N°8 :

Compléter la grille suivante de manière horizontale par des mots correspondants aux définitions suivantes :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	----------	---	---	---	---	---	---	---	---

1			C	O	L	O	N	N	E			
2			L									
3	E	N	R	E	G	I	S	T	R	E	R	
4			P									
5		P	E	R	S	I	S	T	A	N	C	E
6			I									
7			M	A	N	I	P	U	L	E	R	
8			A									
9			I	N	T	E	R	F	A	C	E	
10			R									
11	A	C	C	E	S	S						

(D) : Clé primaire

La clé primaire d'une table est une colonne ou un groupe de colonnes permettant d'identifier de façon unique chaque ligne de la table.

Révissez Votre Bac

Notre site « www.BAC.org.tn » vous donne accès à :

1- Des Examens de baccalauréat

2- Des Devoirs de contrôle et synthèse " Sfax et Autres "

3- Des Cours et des résumés " Facile A comprendre "

4- Des Séries avec corrigés

5- Des Quiz et des tests d'intelligence avec score

6- Des Groupes de discussion privée pour résoudre vos problèmes

7- Vous Pouvez Gagnés D'argent Facilement



EXERCICES

Exercice N° 1 :

Mettre dans la case correspondante la lettre (V) si la réponse est correcte et (F) sinon.

Dans une base de données à plusieurs tables on peut dire que :

<input type="checkbox"/>	Les tables servent à stocker les données
<input type="checkbox"/>	Les tables servent à stocker les données avec des règles de validation
<input type="checkbox"/>	Les tables doivent comporter chacune au moins deux clés étrangères

Une clé primaire :

<input type="checkbox"/>	Doit être définie par une seule colonne
<input type="checkbox"/>	Peut être une clé étrangère dans une autre table
<input type="checkbox"/>	Ne peut être que de type numérique

La création d'une clé primaire dans une table a pour objectif de :

<input type="checkbox"/>	Différencier sans ambiguïté les lignes
<input type="checkbox"/>	Accélérer la recherche des données
<input type="checkbox"/>	Aider à l'élaboration de liens entre les tables

Une table peut comporter

<input type="checkbox"/>	Zéro clé primaire
<input type="checkbox"/>	Une clé primaire
<input type="checkbox"/>	Une ou plusieurs clés primaires

Exercice N° 2 :

1) Compléter les phrases suivantes par les mots de cette liste : «**Redondance, Structuré, tables, relations, la clé primaire, enregistrements, champs, permanentes**».

Une base de données est un ensemble de données enregistré sur des mémoires crée et tenu à jour pour les besoins d'un ensemble d'utilisateurs de cette base. Le SGBD est un outil permettant de créer, de mettre à jour, de rechercher et de consulter les données.

Le SGBD relationnel organise les données sous la forme de Chaque table est une collection de données relatives à un sujet spécifique du monde réel. Ces données sont organisées en colonnes, appelées : attributs ou et en lignes, appelées ou n-uplets.

Un ou plusieurs champs qui permettent d'identifier d'une manière unique les enregistrements, représentent Les tables d'une base de données sont souvent liées par des pour éviter la

2) Compléter les phrases suivantes par les mots de cette liste : «**redondance, enregistrements, données, colonnes, relationnelle, structurée, tables, données, structure, base, champs, lignes**».

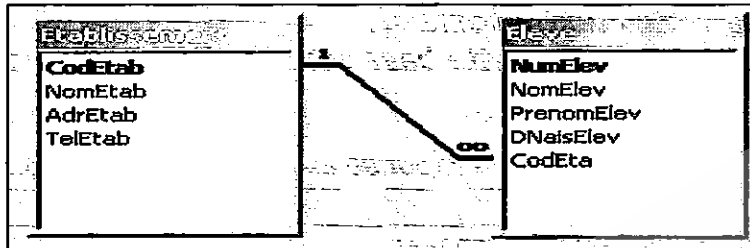
Une base de données est un ensemble de stockées de façon et avec le moins de

Les données d'une de données sont organisées sous forme de

Les données d'une table sont organisées en et en ...: les colonnes appelées souvent définissant la de la table et les lignes appelées souvent constituant les de la table.

Exercice N° 3 :

Soit la représentation graphique suivante :



1) Compléter le vide par un commentaire approprié :

NumElev : est

CodEtab : est

NomElev : est

CodEta : est

Etablissement est

La table mère est la table (Etablissement ou Eleve)

La cardinalité "1" à côté de la table Etablissement veut dire que

2) Donner la représentation textuelle de ces tables.

Exercice N° 4 :

Compléter les cases du tableau suivant :

Terme	Définition
.....	Propriété d'une entité dans un modèle correspondant généralement à une colonne dans une table de la base.
Base de données
.....	Ensemble de données relatives à un même concept.
.....	Modèle de représentation des données sous forme d'arborescence.
.....	Elément vertical dans une table représentant un ensemble de valeurs d'un attribut.
Contrainte d'intégrité
.....	Association entre deux entités, représentée dans l'univers des SGBDR par l'ajout d'une clé étrangère.
SGBD

Exercice N° 5 :

On considère une base de données qui contient des informations sur les avions qui ont participé à une guerre. Un avion a un numéro d'immatriculation, un nom, et on connaît la date de son premier vol. Un avion appartient à une classe. Une classe d'avions possède un identifiant (nom de classe), un poids et la distance franchissable.

On mémorise également les batailles aériennes : leurs numéros, nom, date de début et date de fin.

Enfin lorsqu'un avion a participé à une bataille, on mémorise l'état dans lequel il se trouvait à l'issue de la bataille ("ok", "abîmé", ou "coulé").

N.B. : Un avion peut participer à plusieurs batailles et dans une bataille peut participer plusieurs avions.

1. Représenter les tables textuellement .

2. Représenter la structure de cette base sous forme graphique.

Exercice N° 6 :

Mettre la lettre « V » devant la bonne réponse et la lettre « F » sinon :

1) le SGBD permet :	
	La manipulation des Bases de Données
	La manipulation du moteur
	De ranger les données dans des documents
2) une ligne correspond à :	
	Une clé étrangère
	Une occurrence du sujet représenté par la colonne
	Un champ
3) une contrainte d'intégrité est :	
	Peut être appliquée à un attribut
	Une valeur logique
	Une condition qui doit être toujours vérifiée
4) la clé étrangère :	
	Identifie d'une façon unique l'enregistrement
	Représente le lien existant entre les tables
	Est obligatoire dans chaque table
5) l'intérêt de l'utilisation d'une base de données	
	La mise en place d'un SGBD
	L'organisation des données
	Les données seront plus visibles
6) le langage de manipulation des données permet de :	
	Décrire les objets qui constituent la base
	Décrire les contraintes concernant les objets de la base
	Créer, modifier, supprimer, ...les données de la base
7) le modèle relationnel est un :	
	Graphe
	Lien entre père fils
	Ensemble d'objets représentés par des tables

Exercice N° 7 :

- Soit la représentation textuelle de la BD « Compétition » suivante :

Coureurs (Num-Cou, Nom-cou)

Equipes (Num-Eq, Nom-Eq, Directeur)

Pays (Cod-Pays, Nom-pays)

Étapes (Num-Étap, Date, Ville-Dep, Ville-Arr, NbKm)

Participer (temps-réalisé)

- Soit les hypothèses suivantes :

1) Une équipe est formée de plusieurs coureurs.

2) Un pays peut participer à la compétition par plusieurs coureurs.

3) Un coureur peut participer à plusieurs étapes, et à chaque étape le coureur réalise un temps.

Travail demandé :

1) Suivant les hypothèses indiquées, déterminer la nouvelle représentation textuelle de cette BD en précisant pour chaque table sa clé primaire ainsi que les clés étrangères déduites.

Exercice N° 8 :

Soit une BD « Gestion des pièces » formée par les tables suivantes :

Produits (Num-p, Des-p, couleur, poids)

Commande (Num-cmd, qte-cmd)

Services (Num-s, lib-s, localisation)

Soit les hypothèses suivantes.

① Un produit peut être passé dans plusieurs commandes

② Une commande ne peut être réalisée que par service et par produit

③ Un service peut réaliser plusieurs commandes

→ En tenant compte des hypothèses, réécrire la structure de cette BD, en précisant les clés primaires et étrangères.

Exercice N° 9 :

Soit une BD « Gestion des projets » formée par les tables suivantes :

Employés (Num-emp, nom, prenom, adresse, tél, grade)

Services (Num-s, nom-s, responsable, tél)

Projets (Num-proj, nom-proj, dat-deb, dat-fin)

Soit les deux hypothèses suivantes :

① Un service contient plusieurs employés

② Plusieurs projets peuvent être réalisés par un service

→ En tenant compte des deux hypothèses, réécrire la structure de cette BD en précisant les clés primaires et étrangères

Exercice N° 10 :

Soit la représentation textuelle de la BD « Gestion départ » suivante :

Employés (Matr, nomE, grade, dat-emb, salaire)

Départements (Ndept, Nomdep, lieu)

Projets (Cod-P, Nom-P)

Participations (heures)

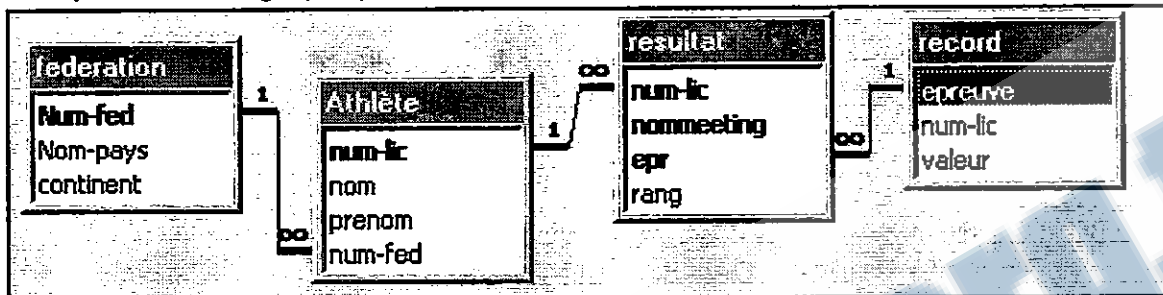
Soit les hypothèses suivantes :

- ① Un département est formé de plusieurs employés.
- ② Un employé peut participer à plusieurs projets, et à chaque projet l'employé met un nombre d'heurs.

→ Suivant les hypothèses indiquées, déterminer la nouvelle représentation textuelle de cette BD en précisant pour chaque table sa clé primaire ainsi que les clés étrangères déduites.

Exercice N° 11 :

Soit la représentation graphique d'une BD ci-dessous :



- Réécrire cette représentation en mode textuel.

CORRECTION

Exercice N° 1 :

V
V
F

F
V
F

V
V
V

F
V
V

Exercice N° 2 :

1) Une base de données est un ensemble **structuré** de données enregistré sur des mémoires **permanentes** crée et tenu à Jour pour les besoins d'un ensemble d'utilisateurs de cette base. Le système de gestion de bases de données est un outil permettant de créer, de mettre à jour, de rechercher et de consulter les données.

Le SGBD relationnel organise les données sous la forme de **tables**. Chaque table est une collection de données relatives à un sujet spécifique du monde réel. Ces données sont organisées en colonnes, appelées : attributs ou **champs** et en lignes, appelées **enregistrements** ou n_uplets.

Un ou plusieurs champs qui permettent d'identifier d'une manière unique les enregistrements, représentent **la clé primaire**. Les tables d'une base de données sont souvent liées par des **relations** pour éviter la **redondance** (Duplication d'une même information dans plusieurs tables).

2) Une base de données est un ensemble de **données** stockées de façon **structurée** et avec le moins de **redondance**.

Les données d'une **base** de données **relationnelle** sont organisées sous forme de **tables**.

Les données d'une table sont organisées en **colonnes** et en **lignes** : les colonnes appelées souvent **champs** définissant la **structure** de ta table et les lignes appelées souvent **enregistrements** constituant les **données** de la table.

Exercice N° 3 :

- 1) NumElev : est **une clé primaire**
 CodEtab : est **une clé primaire**
 NomElev : est **un champ**

CodEta : est **une clé étrangère**

Etablissement est **table**

La table mère est la table (Etablissement ou Eleve) : **Etablissement**

La cardinalité "1" à côté de la table Etablissement veut dire que : Etablissement **est la table mère et CodEta est une clé primaire.**

2) Etablissement (CodEtab, NomEtab, AdrEtab, TelEtab)

Eleve (NumElev, NomElev, PrenomElev, DnaisElev, CodEtab#)

Exercice N° 4 :

Terme	Définition
Attribut	Propriété d'une entité dans un modèle correspondant généralement à une colonne dans une table de la base
Base de données	Ensemble structuré et organisé d'informations permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation
Table	Ensemble de données relatives à un même concept.
Modèle hiérarchique	Modèle de représentation des données sous forme d'arborescence.
Colonne	Élément vertical dans une table représentant un ensemble de valeurs d'un attribut.
Contrainte d'intégrité	Règles qui doivent être toujours vérifiées
Relation	Association entre deux entités, représentée dans l'univers des SGBDR par l'ajout d'une clé étrangère.
SGBD	Ensemble de logiciels permettant de modifier ou d'interroger les données d'une base.

Exercice N° 5 :

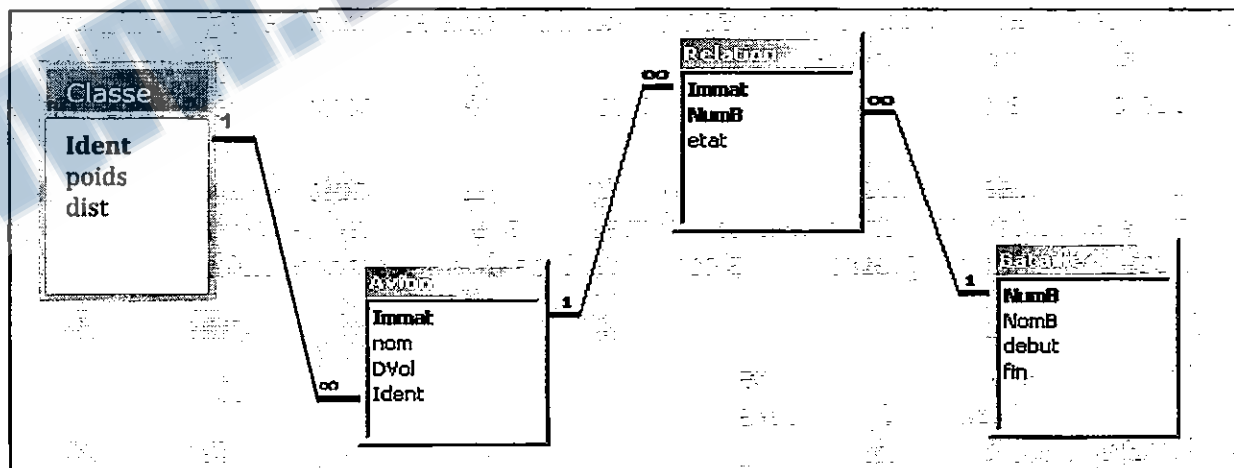
1) Avion (Immat, Nom, D_vol, ident#)

Classe (Ident, poids, dist)

Bataille (NumB, NomB, Debut, fin)

Relation (Immat#, NumB#, etat)

2)



Exercice N° 6 :

Mettre la lettre « V » devant la bonne réponse et la lettre « F » sinon :

V
F
F

F
V
F

V
F
V

F
V
F

F
V
V

F
F
V

F
F
V

Exercice N° 7 :

- Une équipe est formée de plusieurs coureurs : signifie qu'un coureur possède une seule équipe ; donc Num-Eq sera une clé étrangère dans la table Coureurs.
- Un pays peut participer à la compétition par plusieurs coureurs : signifie qu'un coureur possède un seule pays ; donc Cod-Pays sera une clé étrangère dans la table Coureurs.
- Un coureur peut participer à plusieurs étapes, et à chaque étape le coureur réalise un temps : signifie Num-Cou et Num-Etap seront des clés étrangères dans la table Participer et constitueront ensemble la clé primaire de cette table.

→ La structure de la BD sera la suivante :

Coureurs (Num-Cou, Nom-cou, Num-Eq#, Cod-Pays#)

Equipes (Num-Eq, Nom-Eq, Directeur)

Pays (Cod-Pays, Nom-pays)

Etapes (Num-Etap, Date, Ville-Dep, Ville-Arr, NbKm)

Participer (Num-Cou#, Num-Etap#, temps-réalisé)

Exercice N° 8 :

- ① Un produit peut être passé dans plusieurs commandes
- ② Une commande ne peut être réalisée que par service et par produit
- ③ Un service peut réaliser plusieurs commandes

D'après ces hypothèses, on ajoutera à la table **Commande** deux clés étrangères **Num-p** de la table **Produits** et **Num-s** de la table **Services**. Pour identifier une commande, il faut savoir le service et le produit (hypothèse 2) ; donc ces deux clés étrangères seront intégrés dans la clé primaire de la table **Commande** avec **Num-cmd**.

Produits (Num-p, Des-p, couleur, poids)

Commande (Num-cmd, Num-p#, Num-s#, qte-cmd)

Services (Num-s, lib-s, localisation)

Exercice N° 9 :

- ① Un service contient plusieurs employés : signifie que la clé primaire de la table **Services** migrera comme clé étrangère dans la table **Employés**.
- ② Plusieurs projets peuvent être réalisés par un service : signifie que la clé primaire de la table **Services** migrera comme clé étrangère dans la table **Projets**.

Employés (Num-emp, nom, prenom, adresse, tél, grade, Num-s#)

Services (Num-s, nom-s, responsable, tél)

Projets (Num-proj, nom-proj, dat-deb, dat-fin, Num-s#)

Exercice N° 10 :

- ① Un département est formé de plusieurs employés : signifie que la clé primaire de la table **Départements** migrera comme clé étrangère dans la table **Employés**.
- ② Un employé peut participer à plusieurs projets, et à chaque projet l'employé met un nombre d'heurs : on ajoutera à la table **Participations** deux clés étrangères **Matr** de la table **Employés** et **Cod-P** de la table **Projets**. Pour identifier une participation, il faut savoir l'employé et le projet; donc ces deux clés étrangères seront intégrés dans la clé primaire de la table **Participations**.

Employés (Matr, nomE, grade, dat-emb, salaire, Ndept #)

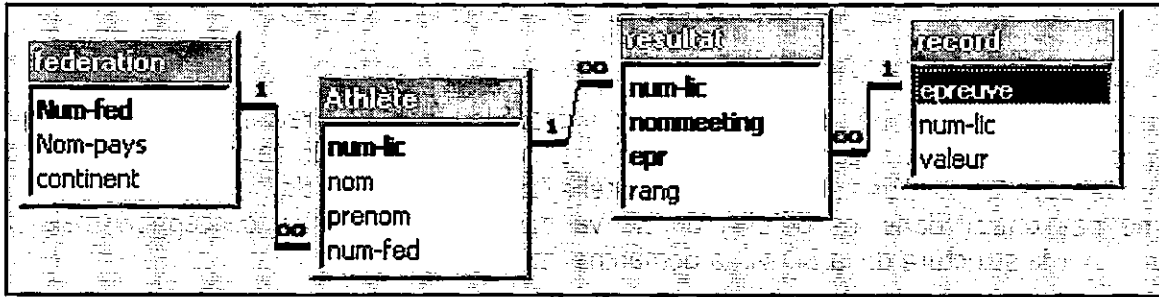
Départements (Ndept, Nomdep, lieu)

Projets (Cod-P, Nom-P)

Participations (Matr#, Cod-P#, heures)

Exercice N° 11 :

Soit la représentation graphique d'une BD ci-dessous :



Athlète (Num-lic, nom, prenom, num-fed#)

federation (Num-fed, Nom-pays, continent)

record (epreuve, num-lic#, valeur)

Participations (Num-lic #, epr#, nommeeting, rang)

Exercices

Exercice n°1 :

www.BAC.org.tn
Page : BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Ecrire un programme, en Turbo Pascal, qui permet de déterminer les coordonnées d'un point S, représentant la symétrie d'un point P dans un plan à deux dimensions.

N.B : Un point P, dans un plan à deux dimensions, est caractérisé par deux coordonnées cartésiennes (x,y) représentant respectivement son abscisse et son ordonnée.

Exemple :

Si P est de coordonnées (2,3) → S sera de coordonnées (-2,-3)

Exercice n°2 :

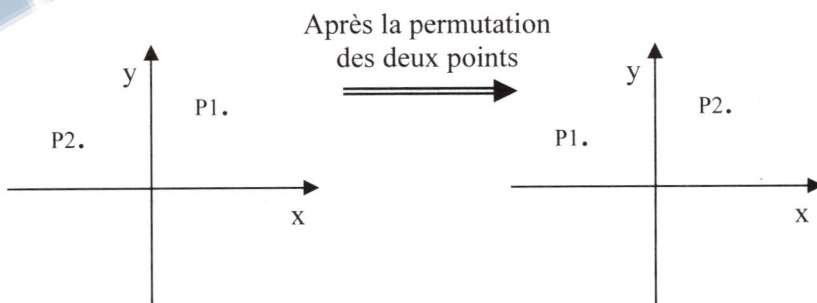
Ecrire un programme, en Turbo Pascal, qui permet de permuter la disposition des deux points P1 et P2 dans un plan à deux dimensions.

Exemple :

Si P1 = (1.5 , 2) et P2 = (-1.5 , 1)

La permutation des coordonnées de deux points donne le résultat suivant :

P1 = (-1.5 , 1) et P2 = (1.5 , 2)



www.BAC.org.tn
Page : BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°3 :

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Un élève est caractérisé par les données suivantes :

- ↗ **Nom** : Chaîne [20], commençant par une lettre en majuscule,
- ↗ **Prénom** : Chaîne [20], commençant par une lettre en majuscule,
- ↗ **Age**: Entier de 6 à 22 ans
- ↗ **Moyenne**: Réel entre 0 et 20

Faire l'analyse et déduire les algorithmes d'un programme qui permet de saisir les données d'un élève et de les afficher.

Exercice n°4 :

La tâche est simple avec un seul élève ! Quelle sera, maintenant la solution si on vous demande de saisir et d'afficher des données d'une classe de n élèves ?

Exercice n°5 :

Faire la traduction, en Turbo pascal, des algorithmes obtenus au niveau de l'exercice n°4 en ajoutant les modules suivants :

Module 1 : Un sous programme permettant d'afficher les noms et les prénoms des élèves ayant une moyenne ≥ 12 .

Module 2 : Un sous programme permettant de déterminer le nombre d'élèves refusés (leurs moyennes < 10).

Exercice n°6 :

Une agence de location d'automobiles dispose de n véhicules, avec $n \in [4..100]$, désire informatiser quelques tâches de gestion de son parc de véhicules.

Chaque véhicule est caractérisé par :

- ↗ Une marque (chaîne de 20cc)
- ↗ Un numéro d'immatriculation (chaîne de 11cc)
- ↗ Une date de de la 1^{ère} mise en circulation de la forme jj/mm/aaaa
- ↗ Un nombre de personnes à transporter (de 1 à 50)
- ↗ Un indice de kilométrage ($\leq 150\ 000$)
- ↗ Une disponibilité = Vrai : si le véhicule est disponible pour la location ou = faux sinon (en panne, déjà loué,...),

www.BAC.org.tn
Page BAC-TUNISIE

Faire l'analyse et déduire les algorithmes d'un programme permettant de :

- ✓ Saisir le nombre de véhicules,
- ✓ Remplir les données de tous les véhicules,
- ✓ Afficher les numéros d'immatriculation des véhicules disponibles et dont le nombre de personnes à transporter = un nombre donné,
- ✓ Calculer le nombre de véhicules qui ont roulé moins d'un kilométrage donné en km,
- ✓ Calculer le nombre de véhicules mis en circulation après une année donnée par l'utilisateur.

Exercice n°7 :

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Revenant à l'exercice n° 5 et 6!

- a) Est-ce que les données saisies, correspondantes aux élèves d'une classe ou aux véhicules de l'agence sont enregistrées sur un support physique de votre ordinateur (disquette, disque dur, ...) ?
- b) Si oui, c'est sur lequel des supports ? Sinon pourquoi ?
- c) Au cas où les données n'ont pas été enregistrées, y-a-t-il une solution pour les sauvegarder sur un support physique ?

Exercice n°8 :

Remplir le tableau ci-dessous pour comparer les tableaux et les fichiers, en utilisant les termes suivants :

Temporaire, Mémoire secondaire, Limitée, Simples et composées, Quasi-permanente, RAM, Illimitée.

	Durabilité	Taille	Support	Type de données
Tableau				
Fichier				

Exercice n°9 :

Refaire le programme Eleves (exercice n°5), en Turbo Pascal, en assurant les mêmes tâches demandées, mais cette fois en stockant les données des n élèves sur le disque dur dans un fichier physique qui porte comme nom « Eleve.DAT ».

www.BAC.org.tn
Page BAC-TUNISIE

Exercice n°10 :

Faire l'analyse et déduire les algorithmes d'un programme qui permet de créer un fichier « Lettre.DAT » sur le disque dur « C : » dont :

- ☞ Le contenu est un ensemble de caractères alphabétiques.
- ☞ Le nombre d'éléments n'est pas connu au préalable, l'arrêt de la saisie est réalisé lorsque l'utilisateur tape '#'.

Le programme doit assurer, alors, les tâches suivantes :

- ✓ Calculer le nombre de voyelles se trouvant dans le fichier,
- ✓ Calculer le nombre de consonnes se trouvant dans le fichier,
- ✓ Mettre les voyelles dans un 1^{er} fichier "Voyelles.DAT" et les consonnes dans un 2^{ème} fichier "Consonnes.DAT",
- ✓ Afficher le contenu de chacun des deux nouveaux fichiers.

Exercice n°11:

Faire l'analyse et déduire les algorithmes d'un programme qui permet de créer un 1^{er} fichier « Resultat.FCH » dont le contenu est relatif au résultat d'un concours.

Le résultat d'un concours est un ensemble d'informations relatives aux candidats qui ont participé et leurs résultats.

Chaque candidat est caractérisé par :

- ✓ **Nom** : chaîne de 20 cc maximum,
- ✓ **Prénom** : chaîne de 20 cc maximum,
- ✓ **Décision** : 1, 2 ou 3 attribuée par le jury du concours avec 1 : Admis, 2 : Ajourné et 3 : refusé.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Le programme doit assurer les traitements suivants:

1. Afficher les noms et les prénoms des candidats admis,
2. Ranger le nom et le prénom (séparés par un espace) de chaque candidat ajourné dans un 2^{ème} fichier texte intitulé « Ajour.TXT ».
3. Afficher le contenu du fichier « Ajour.TXT ».
4. Calculer et afficher le nombre de candidats refusés.

Il faut prendre en considération les conditions suivantes :

- ☞ Le chemin d'accès des deux fichiers est « C:\Concours\Result »
- ☞ Le nombre d'éléments n'est pas connu au préalable, la continuation ou l'arrêt de la saisie sont laissés au libre choix de l'utilisateur.

Exercice n°12 :

Ecrire un programme, en Turbo Pascal, qui permet de créer un fichier texte "Phrases.TXT" sur la partition "C:".

Le fichier doit contenir des chaînes de caractères, qui commencent par une lettre et se terminent par un point, le nombre de chaînes de caractères est choisi par l'utilisateur.

Le programme doit assurer aussi :

- ↪ L'élimination des espaces superflus dans chaque phrase du fichier : il ne laisse qu'un seul espace entre deux mots de la phrase.
- ↪ L'affichage de la phrase contenant le nombre maximal de mots.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°13:

Faire l'analyse, déduire les algorithmes et faire la traduction, en Turbo Pascal, d'un programme qui permet de créer sur la partition "C:" le fichier "MOTS.TXT" contenant une série de 100 mots au maximum (longueur maximale d'un mot : 30 caractères).

La saisie des mots se terminera à l'introduction du symbole "*" qui ne sera pas écrit dans le fichier.

Le programme affiche, aussi, le nombre de mots, le nombre de palindromes ainsi que la moyenne arithmétique de l'ensemble des caractères se trouvant dans le fichier "MOTS.TXT".

N.B :

- ✓ Un mot ne doit pas contenir un espace,
- ✓ Une chaîne est un palindrome si elle se lit de gauche à droite comme de droite à gauche.

Exemples : "ESSE", "AZIZA", "10011001", etc.

Exercice n°14 :

Ecrire un programme, en Turbo Pascal, qui crée sur la partition "C:" le fichier "Clients.DAT" contenant les données relatives aux comptes des clients de "Essentiel Banque".

Un compte client est caractérisé par :

- **Un numéro de compte** : Entier positif, attribué automatiquement par l'ordinateur,
- **Un Nom** : Chaîne de 20 cc au maximum,
- **Un Prénom** : Chaîne de 20 cc au maximum,
- **Un solde** : Réel.

Le programme assure les requêtes suivantes :

1. L'ajout d'un nouveau client à la fin du fichier,
2. La suppression d'un client choisi par son numéro de compte,
3. La modification du solde d'un client,
4. L'affichage des données d'un client suite à la saisie de son numéro de compte avec l'affichage de l'expression « Débiteur » pour celui qui a un solde < 0 ou « Créiteur » dans le cas contraire.
5. L'affichage des données de tous les clients.

N.B :

- ✓ La **création** ne doit être réalisée que si le fichier est **inexistant!**
- ✓ Il est préféré d'utiliser un **menu** laissant le choix à l'utilisateur d'exécuter la tâche dont il aura besoin autant de fois qu'il le souhaite.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Begin

Saisie (p);
 Sym (p, s);
 Affiche (s);

p et s sont des paramètres effectifs de type enregistrement.

End.**Exercice n°2 :****Program** Permutation;

Uses WinCRT;

TypePoint = **Record**

x, y : Real;

end;

Var

P1, P2 : Point;

Procedure Saisie (Var a : Point);**Begin**

With a do

Begin

Write ('Saisir l'abscisse : ');

Readln (x);

Write ('Saisir l'ordonnée : ');

Readln (y);

end;

End;**Procedure** Permut (Var a, b : Point);**Var**

c : Point;

Begin

c := a;

a := b;

b := c;

End;

L'affectation d'un enregistrement dans un autre est permise à condition que les deux soient de même type. On n'est pas obligé de passer par les champs, comme le cas de la lecture ou de l'affichage des champs d'un enregistrement.

Procedure Affiche (a : Point ; n : byte);**Begin**

Writeln ('P', n, '(', a.x:2:2, ',', a.y:2:2, ')');

End;

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53-371 502

Begin

Writeln ('Saisie des coordonnées du 1er point : ');

Saisie (P1);

Writeln ('Saisie des coordonnées du 2ème point : ');

Saisie (P2);

Permut (P1, P2);

Affiche (P1 , 1);

Affiche (P2 , 2);

End.



Exercice n°3 :

I- Découpage

- ✓ **Eleve** : programme principal, qui permet de saisir et d'afficher les données d'un élève.
- ✓ **Saisie** : procédure, permettant de saisir les données d'un élève.
- ✓ **Affiche** : procédure, permettant d'afficher les données d'un élève.

II- Analyses & Algorithmes

*** Programme principal : Eleve ***

1) Spécification du problème :

Résultat = Affiche (X)

Traitements =

- ✓ X = Saisie (X)
- ✓ Saisie et Affiche = Modules

T.D.O.U.G

Objet	Nature/Type	Rôle
X	Info	Contenir les données d'un élève.
Affiche	Procédure	Afficher les données d'un élève.
Saisie	Procédure	Saisir les données d'un élève.

T.D.N.T

Type
Info = Enregistrement
Nom, pren : Chaîne [20]
Age : 6..22
Moy : Réel
Fin Info

2) Algorithme :

- 0) Début Eleve
- 1) Saisie (X)
- 2) Affiche (X)
- 3) Fin Eleve

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

***** Procédure : Saisie *******1) Spécification du problème :****Résultat = X****Traitements =**

- ✓ X = Données
- ↪ X.Nom = Donnée avec le 1^{er} caractère est une lettre en majuscule,
- ↪ X.Pren = Donnée avec le 1^{er} caractère est une lettre en majuscule,
- ↪ X.Age = Donnée avec $X.Age \in [6..22]$,
- ↪ X.Moy = Donnée avec $X.Moy \in [0..20]$.
- ✓ X = Paramètre formel.

2) Algorithme :

- 0) Procédure Saisie (VAR X : Info)
- 1) Avec X Faire
 - Répéter
 - Ecrire ("Nom : ")
 - Lire (Nom)
 - Jusqu'à Nom[1] Dans ["A".."Z"]
 - Répéter
 - Ecrire ("Prénom : ")
 - Lire (Pren)
 - Jusqu'à Pren[1] Dans ["A".."Z"]

- Répéter
 - Ecrire ("Age : ")
 - Lire (Age)
- Jusqu'à Age Dans [6..22]
- Répéter
 - Ecrire ("Moyenne : ")
 - Lire (Moy)
- Jusqu'à (Moy ≥ 0) ET (Moy ≤ 20)
- FinAvec
- 2) Fin Saisie



***** Procédure : Affiche *****

1) Spécification du problème :

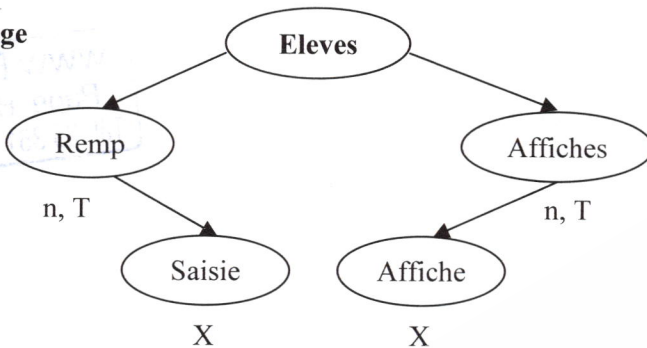
Résultat = Aff

Traitements =

- ✓ Aff = []
 - ↪ Afficher (X.Nom)
 - ↪ Afficher (X.Pren)
 - ↪ Afficher (X.Age)
 - ↪ Afficher (X.Moy)
- ✓ X = Paramètre formel

2) Algorithme :

- 0) Procédure Affiche (X: Info)
- 1) Avec X Faire
 - Ecrire ("Nom : ", Nom)
 - Ecrire ("Prénom : ", Pren)
 - Ecrire ("Age : ", Age)
 - Ecrire ("Moyenne : ", Moy)
- FinAvec
- 2) Fin Affiche

Exercice n°4 :**I- Découpage**

- ✓ **Elevés** : programme principal, qui permet de saisir et d'afficher les données d'une classe de n élèves.
- ✓ **Remp** : procédure, permettant de saisir n et remplir T par les données de n élèves.
- ✓ **Saisie** : procédure, permettant de saisir les données d'un élève.
- ✓ **Affiches** : procédure, permettant d'afficher les données de n élèves.
- ✓ **Affiche** : procédure, permettant d'afficher les données d'un élève.

II- Analyses & Algorithmes

*** Programme principal : Elevés ***

1) Spécification du problème :

Résultat = Affiches (n, T)

Traitements =

- ✓ $n, T = \text{Remp}(n, T)$
- ✓ Remp et Affiches = Modules

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
n	Entier	Contenir le nombre d'élèves d'une classe.
T	TAB	Contenir les données de n élèves.
Affiches	Procédure	Afficher les données de n élèves.
Remp	Procédure	Saisir n et remplir T .

T.D.N.T

Type
TAB = Tableau de 40 Info
Info = Enregistrement
Nom, Pren : Chaîne [20]
Age : 6..22
Moy : Réel
Fin Info

2) Algorithme :

- 0) Début Eleves
- 1) Remp (n, T)
- 2) Affiches (n, T)
- 3) Fin Eleves

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Remp ***

1) Spécification du problème :

Résultat = n et T

Traitements =

- ✓ n = Donnée avec $n \in [1..40]$
- ✓ $T = [\quad]$
- ✎ $T[i] = \text{Saisie}(T[i])$
- ✓ i = Compteur allant de 1 à n
- ✓ Saisie = Procédure
- ✓ n et T = Paramètres formels

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
Saisie	Procédure	Saisir les données d'un élève.
i	Entier	Compteur.

2) **Algorithme :**

0) Procédure Remp (VAR n : Entier ; VAR T : TAB)

1) Répéter

Ecrire ("Le nombre d'élèves ? : ")

Lire (n)

Jusqu'à n Dans [1..40]

2) Pour i de 1 à n Faire

Saisie (T[i])

FinPour

3) Fin Remp

*** Procédure : Saisie ***

La même démarche que la procédure Saisie de l'exercice n°3 page 24

*** Procédure : Affiches ***

1) **Spécification du problème :****Résultat** = Aff**Traitements** =

✓ Aff = []

↳ Affiche (T[i])

✓ i = Compteur allant de 1 à n

✓ Affiche = Procédure

✓ n, T = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
Affiche	Procédure	Afficher les données d'un élève.
i	Entier	Compteur.

2) **Algorithme :**

0) Procédure Affiches (X: Info)

1) Pour i de 1 à n Faire

Affiche (T[i])

FinPour

2) Fin Affiches

*** Procédure : Affiche ***

La même démarche que la procédure Affiche de l'exercice n°3 page 25

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°5 :**Program** Eleves;

Uses winct;

TypeInfo = **Record**

Nom, Pren : String [20];

Age : 6..22;

Moy : Real;

End;

Tab = Array [1..40] of Info;

Var

n : integer;

t : tab;

Procedure Saisie (Var x : Info);**Begin**

With x do

Begin

Repeat

Write ('Nom : ');

Readln (Nom);

Until Nom[1] in ['A'..'Z'];

Repeat

Write ('Prénom : ');

Readln (Pren);

Until Pren [1] in ['A'..'Z'] ;

Repeat

Write ('Age : ');

Readln (Age);

Until Age in [6..22];

Repeat

Write ('Moyenne : ');

Readln (Moy);

Until (Moy>=0) And (Moy<=20);

end;**End;**

```
Procedure Remp (Var n:integer ; Var T: Tab);
```

```
Var i:integer;
```

```
Begin
```

```
  Repeat
```

```
    Write ('Le nombre d"élèves ? : ');
```

```
    Readln (n);
```

```
  Until n in [1..40];
```

```
  For i:= 1 to n do
```

```
  begin
```

```
    Writeln ('Elève n ° ,i, ' : ');
```

```
    Saisie (T[i]);
```

```
  end;
```

```
End;
```

```
Procedure Affiche ( x : Info);
```

```
Begin
```

```
  With x do
```

```
  Begin
```

```
    Writeln ('Nom = ', Nom);
```

```
    Writeln ('Prénom = ', Pren);
```

```
    Writeln ('Age = ', Age);
```

```
    Writeln ('Moyenne = ', Moy:2:2);
```

```
  end;
```

```
Writeln;
```

```
End;
```

```
Procedure Affiches (n : integer ; T : Tab);
```

```
Var
```

```
  i : integer;
```

```
Begin
```

```
  for i:= 1 to n do
```

```
    Affiche (T[i]);
```

```
End;
```

```
Procedure Sup12 (n : Integer; T:Tab);
```

```
Var
```

```
  i : integer;
```

```
Begin
```

```
  For i:= 1 to n do
```

```
  With T[i] do
```

```
    If Moy >= 12 Then Writeln (Nom, ' ', Pren);
```

```
End;
```

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Function Inf10 (n : Integer; T:Tab): Integer;

Var

i,x : integer;

Begin

x:=0;

For i:= 1 to n do

With T[i] do

If Moy < 10 Then Inc(X);

Inf10:= x;

End;

{Programme principal}

Begin

Writeln ('*** Remplissage des données des élèves ***');

Remp (n, T);

Writeln;

Writeln ('** Affichage de tous les élèves **');

Affiches (n, T);

Writeln ('** Affichage des élèves ayant une moyenne >= 12 **');

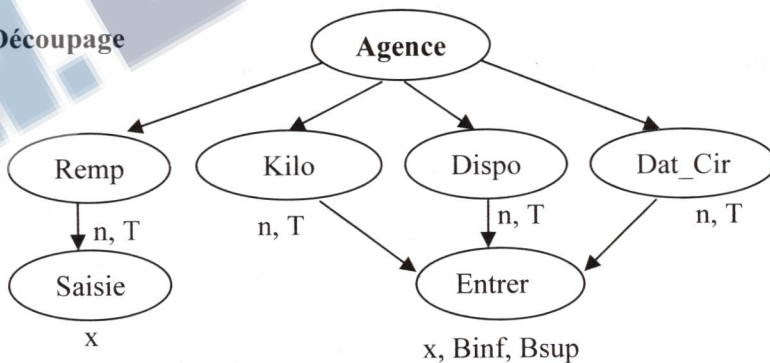
Sup12 (n, T);

Writeln ('Le nombre d"élèves refusés =', Inf10 (n, T));

End.

Exercice n°6 :

I- Découpage



- ✓ **Agence** : programme principal, qui permet de saisir les données des n véhicules et de répondre à certaines requêtes de l'utilisateur.

T.D.N.T

Type	
TAB = Tableau de 100 VEH	
VEH = Enregistrement	
MRQ : Chaîne [20]	
NUM : Chaîne [11]	
DMC : Date ←	DMC est un champ de type Date qui est un nouveau type, c'est un enregistrement dans un enregistrement.
NPT : 1..50	
IND : 0..150000	
DIS : Booléen	
Fin VEH	
Date = Enregistrement	
JJ : 1..31	
MM : 1..12	
AN : 2007 .. 2999	
Fin Date	

2) Algorithme :

- 0) Début Agence
- 1) Remp (n, T)
- 2) Kilo (n, T)
- 3) Dat_Cir (n, T)
- 4) Dispo (n, T)
- 5) Fin Agence

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

***** Procédure : Remp *****

1) Spécification du problème :

Résultat = n et T

Traitements =

- ✓ n = Donnée avec $n \in [4..100]$
- ✓ T = []
- ↪ T[i] = Saisie (T[i])
- ✓ i = Compteur allant de 1 à n
- ✓ Saisie = Procédure
- ✓ n et T = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
Saisie	Procédure	Saisir les données d'un véhicule.
i	Entier	Compteur.

2) Algorithme :

0) Procédure Remp (VAR n : Entier ; VAR T : TAB)

1) Répéter

Ecrire ("Le nombre de véhicules ? :")

Lire (n)

Jusqu'à n Dans [4..100]

2) Pour i de 1 à n Faire

Saisie (T[i])

FinPour

3) Fin Remp

www.BAC.org.tn

Page: BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

*** Procédure : Saisie ***

1) Spécification du problème :

Résultat = X

Traitements =

✓ X = Données

↗ X.MRQ = Donnée avec le 1^{er} caractère est une lettre en majuscule,

↗ X.NUM = Donnée avec X.NUM est ≠ du vide,

↗ X.DMC = Données

▪ X.DMC.JJ = Donnée avec X.DMC.JJ ∈ [1..31],

▪ X.DMC.MM = Donnée avec X.DMC.MM ∈ [1..12],

▪ X.DMC.AN = Donnée avec X.DMC.AN ∈ [2007..2999].

↗ X.NPT = Donnée avec X.NPT ∈ [1..50],

↗ X.IND = Donnée avec X.IND ∈ [0..150 000].

↗ X.DIS = Vrai si Choix = 1,

= Faux si Choix = 2,

↗ Choix = Donnée avec Choix ∈ [1,2],

✓ X = Paramètre formel.

T.D.O.U.L

Objet	Nature / Type	Rôle
Choix	1..2	Contenir un choix de l'utilisateur.

2) Algorithme :

0) Procédure Saisie (VAR X : Info)

1) **Avec X Faire**

Répéter

Ecrire ("Marque : ")

Lire (MRQ)

Jusqu'à MRQ[1] Dans ["A".."Z"]

Répéter

Ecrire ("Numéro d'immatriculation : ")

Lire (NUM)

Jusqu'à NUM<>""

Ecrire ("Date de la 1ère mise en circulation : ")

Avec DMC Faire

Répéter

Ecrire ("Jour : ")

Lire (JJ)

Jusqu'à JJ Dans [1..31]

Répéter

Ecrire ("Mois : ")

Lire (MM)

Jusqu'à MM Dans [1..12]

Répéter

Ecrire ("An : ")

Lire (AN)

Jusqu'à (AN >=2007) ET (AN <=2999)

Fin Avec

Répéter

Ecrire ("Nombre de personnes à transporter : ")

Lire (NPT)

Jusqu'à NPT Dans [1..50]

Répéter

Ecrire ("Indexe de kilométrage : ")

Lire (IND)

Jusqu'à (IND >=0) ET (IND <=150000)

Répéter

Ecrire ("1- Disponible *** 2- Indisponible ")

www.BAC.org.tn

Page: BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

Ecrire ("Tapez votre choix : ") ; Lire (Choix)
 Jusqu'à Choix Dans [1,2]
 Si Choix = 1 Alors DIS ← Vrai Sinon DIS ← Faux Finsi
FinAvec

2) Fin Saisie

*** Procédure : Entrer ***

1) Spécification du problème :

Résultat = x

Traitements =

- ✓ x = Donnée avec $x \in [\text{Binf} .. \text{Bsup}]$
- ✓ x, Binf et Bsup = Paramètres formels

2) Algorithme :

0) Procédure Entrer (Var x: EntierLong ; Binf, Bsup: EntierLong)

1) Répéter

Lire (x)

Jusqu'à (x >= Binf) ET (x <= Bsup)

2) Fin Entrer

*** Procédure : Kilo ***

1) Spécification du problème :

Résultat = Afficher (nb)

Traitements =

- ✓ nb = [nb ← 0]
- ✎ nb ↗ +1 Si T[i].IND ≤ K
- ✓ i = Compteur allant de 1 à n
- ✓ K = Entrer (K, 0, 150000)
- ✓ Entrer = Procédure (partagée avec : Dispo et Dat_Cir)
- ✓ n et T = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
nb	Entier	Contenir le nombre de véhicules qui ont roulés moins de K Km.
K	EntierLong	Contenir une mesure kilométrique.
i	Entier	Compteur.

2) Algorithme :

0) Procédure Kilo (n : Entier ; T : TAB)

1) Ecrire ("Kilométrage minimal? : ")
Entrer (K, 0, 150000)

2) nb ← 0

Pour i de 1 à n Faire

Si T[i].IND ≤ K

Alors

nb ← nb + 1

FinSi

FinPour

3) Ecrire ("Le nombre de véhicules qui ont roulés moins de ", K,
" km = ", nb)

4) Fin Kilo

***** Procédure : Dispo *******1) Spécification du problème :****Résultat** = Afficher (nb)**Traitements** =

✓ nb = [nb ← 0]

↳ nb ↗ + 1 Si T[i].NPT = P et T[i].DIS = Vrai (véhicule disponible)

✓ i = Compteur allant de 1 à n

✓ P = Entrer (P, 1, 50)

✓ Entrer = Procédure (partagée avec : kilo et Dat_Cir)

✓ n et T = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
nb	Entier	Contenir le nombre de véhicules disponibles et dont le nombre de personnes à transporter = P.
P	EntierLong	Contenir le nombre de personnes à transporter.
i	Entier	Compteur.

2) Algorithme :

0) Procédure Dispo (n : Entier ; T : TAB)

1) Ecrire ("Nombre de personnes à transporter ? : ")
Entrer (P, 1, 50)

- 2) $nb \leftarrow 0$
 Pour i de 1 à n Faire
 Si $(T[i].NPT = P)$ ET $(T[i].DIS)$
 Alors
 $nb \leftarrow nb+1$
 FinSi
 FinPour
- 3) Ecrire ("Le nombre de véhicules disponibles = ", nb)
 4) Fin Dispo

*** Procédure : Dat_Cir ***

1) Spécification du problème :

Résultat = Afficher (nb)

Traitements =

- ✓ $nb = [nb \leftarrow 0]$
 ↳ $nb \leftarrow +1$ Si $T[i].DMC.AN \geq A$
- ✓ $i =$ Compteur allant de 1 à n
- ✓ $A =$ Entrer ($A, 2007, 2999$)
- ✓ Entrer = procédure (partagée avec : kilo et Dispo)
- ✓ n et $T =$ Paramètres formels

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
nb	Entier	Contenir le nombre de véhicules mis en circulation avant une année = A .
A	Entier Long	Contenir une année entre 2007 et 2999.
i	Entier	Compteur.

- 2) Algorithme :
- 0) Procédure Dat_Cir (n : Entier ; T : TAB)
- 1) Ecrire ("Année de la recherche ? : ")
 Entrer ($A, 2007, 2999$)
- 2) $nb \leftarrow 0$
 Pour i de 1 à n Faire
 Si $T[i].DMC.AN \geq A$ Alors $nb \leftarrow nb+1$ finSi
 FinPour
- 3) Ecrire ("Le nombre de véhicules circulant après cette année = ", nb)
 4) Fin Dat_Cir

Exercice n°7 :

- a) Non, malheureusement, toutes les données saisies, celles des élèves et celles des véhicules, ne sont enregistrées sur aucun support physique. A chaque fois où l'on exécute l'un des deux programmes on se trouve de nouveau dans l'obligation de ressaisir les données afin de les traiter.
- b) La perte des données saisies est due à la structure de donnée dans laquelle le programme les met ! Les deux programmes utilisent le tableau T comme structure de données pour traiter les données des élèves ou des véhicules. Cette structure (tableau) est un ensemble de cases mémoires de la R.A.M. et cette dernière est une mémoire temporaire, dynamique et volatile, l'exécution d'un programme engendre l'initialisation des cases mémoire allouées de ladite R.A.M.
- c) Afin de sauvegarder ces données sur un support physique la solution est d'utiliser **les fichiers**.

Exercice n°8 :

Le tableau ci-dessous est conçu pour comparer les tableaux et les fichiers :

	Durabilité	Taille	Support	Type de données
Tableau	Temporaire	Limitée	RAM	Simple et composées
Fichier	Quasi-permanente	Illimitée	Mémoire secondaire	Simple et composées

Exercice n°9 :

Program Eleves;

Uses WinCRT;

Type

Info = **Record**

Nom, Pren : String [20];

Age : 6..22;

Moy : Real;

End;

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Fich = **File of** Info;

Var

f : Fich;

Procedure Saisie (Var x : Info);

Begin

With x do

Begin

Repeat

Write ('Nom : ');

Readln (Nom);

Until Nom[1] in ['A'..'Z'];

Repeat

Write ('Prénom : ');

Readln (Pren);

Until Pren [1] in ['A'..'Z'] ;

Repeat

Write ('Age : ');

Readln (Age);

Until Age in [6..22];

Repeat

Write ('Moyenne : ');

Readln (Moy);

Until (Moy>=0) And (Moy<=20);

end;

End;

Procedure Affiche (x : Info);

Begin

With x do

Begin

Writeln ('Nom = ', Nom);

Writeln ('Prénom = ', Pren);

Writeln ('Age = ', Age);

Writeln ('Moyenne = ', Moy;2:2);

end;

Writeln;

End;

Procedure Remp (VAR f : Fich);

Var

i, n : Integer;

x : **Info**;

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Begin

Rewrite (f);

Writeln ('*** Remplissage des données des élèves ***');

Repeat

Write ('Le nombre d"élèves ? : ');

Readln (n);

Until n in [1..40];

For i:= 1 to n do

begin

Writeln ('Elève n ° ',i, ' : ');

Saisie (x);

Write (f, x);

end;

End;

Procedure Affiches (Var f : Fich);

Var

x : Info;

Begin

Reset(f);

Writeln ('*** Liste de tous les élèves ***');

While Not(EOF(f)) do

begin

Read (f, x);

Affiche (x);

end;

End;

Procedure Sup12 (Var f : Fich);

Var

x : Info;

Begin

Reset(f);

Writeln ('*** Liste des élèves ayant une moyenne ≥ 12 ***');

While Not(EOF(f)) do

begin

Read (f, x);

With x do

If Moy ≥ 12

Then Writeln (Nom, ' ', Pren);

end;

End;

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Procedure Inf10 (Var f:Fich);

Var

nb : integer;

x : Info;

Begin

Reset(f);

nb:=0;

While Not(EOF(f)) do

Begin

Read (f, x);

With x do

If Moy < 10

Then Inc (nb);

end;

Writeln ('Le nombre d"élèves refusés = ', nb);

End;

Begin

Assign (f, 'C:\Eleve.DAT');

Remp (f);

Writeln;

Affiches (f);

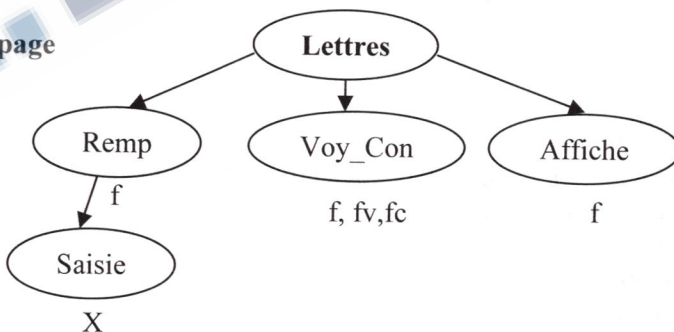
Sup12 (f);

Inf10 (f);

End.

Exercice n°10 :

I- Découpage



- ✓ **Lettres** : programme principal, qui permet de saisir des lettres et de les sauvegarder dans le fichier f.
- ✓ **Remp** : procédure, permettant de créer un fichier de lettres.

- ✓ **Saisie** : procédure, permettant de saisir une lettre ou un "#".
- ✓ **Voy_Con** : procédure, permettant d'éclater le fichier **f** en deux fichiers **fv** fichier des voyelles et **fc** celui des consonnes et d'afficher la taille de chacun des deux fichiers.
- ✓ **Affiche** : procédure, permettant d'afficher le contenu d'un fichier.

II- Analyses & Algorithmes

*** Programme principal : Lettres ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
 - ↪ Affiche (fv)
 - ↪ Affiche (fc)
 - ↪ Afficher (Taille_fichier (fv), Taille_fichier (fc))
- ✓ fv, fc = Voy_Con (f, fv, fc)
- ✓ fv = Assigner (fv, "C:\Voyelles.DAT")
- ✓ fc = Assigner (fc, "C:\Consonnes.DAT")
- ✓ f = Remp (f)
- ✓ f = Assigner (f, "C:\Lettre.DAT")
- ✓ Remp, Voy_Con et Affiche = Modules

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
f	Fich	Contenir des lettres.
fv	Fich	Contenir des voyelles.
fc	Fich	Contenir des consonnes.
Voy_Con	procédure	Eclater le fichier f en deux fichiers fv et fc.
Affiche	Procédure	Afficher le contenu d'un fichier.
Remp	Procédure	Remplir un fichier f par des lettres.

T.D.N.T

Type
Fich = Fichier de caractères

2) Algorithme :

0) Début Lettres

1) Assigner (f, "C:\Lettre.DAT")

Assigner (fv, "C:\Voyelles.DAT")

Assigner (fc, "C:\Consonnes.DAT")

2) Remp (f)

3) Voy_Con(f, fv, fc)

4) Affiche (fv)

Ecrire ("Le nombre de voyelles = ", Taille_fichier (fv))

Affiche (fc)

Ecrire ("Le nombre de consonnes = ", Taille_fichier (fc))

5) Fin Lettres

***** Procédure : Remp *******1) Spécification du problème :****Résultat = f****Traitements =**

✓ f = []

↳ Ecrire (f, x) Si x ≠ "#"

↳ x = Saisie (x)

↳ Répéter ces deux dernières tâches jusqu'à x = "#"

✓ f = Recréer (f)

✓ f = Fermer (f)

✓ f = Paramètre formel

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
x	caractère	Contenir un caractère.

2) Algorithme :

0) Procédure Remp (VAR f: Fich)

1) Recréer (f)

2) Répéter

Saisie (x)

Si x < ">"

Alors Ecrire (f, x)

FinSi

Jusqu'à x = "#"

- 3) Fermer (f)
- 4) Fin Remp

Fermer (Nom_fichier) : c'est une procédure optionnelle, le fichier sera fermé systématiquement à la fin de la procédure.

*** Procédure : Saisie ***

1) Spécification du problème :

Résultat = x

Traitements =

- ✓ x = Donnée avec $x \in ["A" .. "Z", "a" .. "z", "#"]$
- ✓ x = Paramètre formel

2) Algorithme :

0) Procédure Saisie (Var x: Caractère)

1) Répéter

Ecrire ("Saisir une lettre : ")

Lire (x)

Jusqu'à Majus(x) Dans ["A" .. "Z", "#"]

2) Fin Saisie

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Affiche ***

1) Spécification du problème :

Résultat = Aff

Traitements =

✓ Aff = []

a) Afficher (x)

b) x = Lire (f, x)

↳ Répéter les tâches a et b tant qu'il y a des éléments dans le fichier f

✓ f = Ouvrir (f)

✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Caractère	Variable intermédiaire.

2) Algorithme :

- 0) Procédure Affiche (Var f : Fich)
- 1) Ouvrir (f)
- 2) Tantque Non(FF(f)) Faire
 - Lire (f, x)
 - Ecrire (x)
 - FinTantque
- 3) Fin Affiche

*** Procédure : Voy_Con ***

1) Spécification du problème :

Résultat = fv, fc

Traitements =

- ✓ fv, fc = []
 - a) Ecrire (fv, x) si x est une voyelle
 - b) Ecrire (fc, x) si x est une consonne
 - c) x = Lire (f,x)
- ↳ Répéter les tâches a, b et c tant qu'il y a des éléments dans f
- ✓ fv = Recréer (fv)
- ✓ fc = Recréer (fc)
- ✓ f = Ouvrir (f)
- ✓ f, fv et fc = Paramètres formels

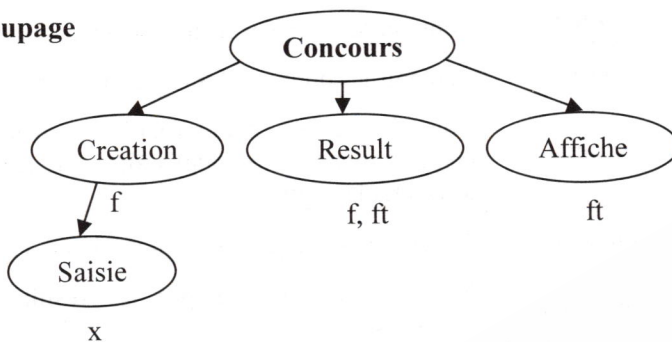
T.D.O.U.L

Objet	Nature / Type	Rôle
x	Caractère	Variable intermédiaire.

2) Algorithme :

- 0) Procédure Voy_Con (Var f, fv, fc : Fich)
- 1) Ouvrir (f)
- 2) Recréer (fv)
- 3) Recréer (fc)
- 4) Tantque Non(FF(f)) Faire
 - Lire (f, x)
 - Si Majus(x) Dans ["A", "E", "U", "I", "Y", "O"]
 - Alors Ecrire (fv, x)
 - Sinon Ecrire (fc, x)
 - FinSi
 - FinTantque
- 5) Fin Voy_Con

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Exercice n°11:**I- Découpage**

- ✓ **Concours** : programme principal, qui permet de gérer les résultats d'un concours.
- ✓ **Creation** : procédure, permettant de créer un fichier contenant les résultats d'un concours.
- ✓ **Saisie** : procédure, permettant de saisir les données d'un candidat.
- ✓ **Result** : procédure, permettant de gérer le résultat du concours.
- ✓ **Affiche** : procédure, permettant d'afficher le contenu d'un fichier.

II- Analyses & Algorithmes

*** Programme principal : Concours ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
 - ↳ Affiche (ft)
 - ↳ Result (f,ft)
- ✓ ft = Créé et rempli dans la procédure Result
- ✓ ft = Assigner (ft, "C:\Concours\Result\Ajour.TXT")
- ✓ f = Creation (f)
- ✓ f = Assigner (f, " C:\Concours\Result\Resultat.FCH")
- ✓ Creation, Result et Affiche = Modules

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
f	Fich	Contenir les données d'un concours.
ft	Texte	Contenir les données des candidats ajournés.
Result	procédure	Créer le fichier "Ajour.TXT" et afficher les noms et prénoms des admis ainsi que le nombre des candidats refusés.
Affiche	Procédure	Afficher le contenu du fichier "Ajour.TXT".
Creation	Procédure	Remplir un fichier f par les résultats d'un concours.

T.D.N.T

Type
Fich = Fichier de Cand
Cand = Enregistrement
Nom, Pren : Chaîne [20]
D : 1..3
Fin Cand

2) Algorithme :

- 0) Début Concours
- 1) Assigner (f, "C:\Concours\Result\Resultat.FCH")
Assigner (ft, "C:\Concours\Result\Ajour.TXT")
- 2) Creation (f)
- 3) Result (f, ft)
Affiche (ft)
- 4) Fin Concours

*** Procédure : Creation ***

1) Spécification du problème :

Résultat = f

Traitements =

✓ f = []

- a) Ecrire (f, x)
- b) x = Saisie (x)

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

- ↪ Répéter les deux tâches a et b jusqu'à Majus (Rep) = "N"
- ↪ Rep = Donnée avec Majus (Rep) ∈ ["O", "N"]
- ✓ f = Recréer (f)
- ✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
Rep	Caractère	Contenir la réponse de l'utilisateur.
x	Cand	Variable intermédiaire.

2) Algorithme :

0) Procédure Creation (VAR f: Fich)

1) Recréer (f)

2) **Répéter**

Saisie (x)

Ecrire (f,x)

Répéter

Ecrire ("Autre candidat (O/N) ? : ")

Lire (Rep)

Jusqu'à Majus (Rep) Dans ["O", "N"]**Jusqu'à** Majus (Rep) = "N"

3) Fin Creation

***** Procédure : Saisie *******1) Spécification du problème :****Résultat** = x**Traitements** =

- ✓ x = Données
- ↪ x.Nom = Donnée avec le 1^{er} caractère est une lettre en majuscule,
- ↪ x.Pren = Donnée avec le 1^{er} caractère est une lettre en majuscule,
- ↪ x.D = Donnée avec x.D ∈ [1..3],
- ✓ x = Paramètre formel.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

2) Algorithme :

0) Procédure Saisie (VAR X : Cand)

1) Avec X Faire

Répéter

Ecrire ("Nom : ")

Lire (Nom)

Jusqu'à Nom[1] Dans ["A".."Z"]

Répéter

Ecrire ("Prenom : ")

Lire (Pren)

Jusqu'à Pren[1] Dans ["A".."Z"]

Répéter

Ecrire ("Décision : ")

Lire (D)

Jusqu'à D Dans [1..3]

FinAvec

2) Fin Saisie

*** Procédure : Affiche ***

1) Spécification du problème :

Résultat = Aff

Traitements =

✓ Aff = []

a) Afficher (x)

b) x = Lire_nl (ft, x)

↻ Répéter les tâches a et b tant qu'il y a des lignes dans le fichier ft

✓ ft = Ouvrir (ft)

✓ ft = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Chaîne de caractères	Variable intermédiaire.

2) Algorithme :

0) Procédure Affiche (Var ft : Texte)

1) Ouvrir (ft)

2) Ecrire ("La liste des candidats ajournés : ")

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Tantque Non(FF(ft)) Faire

 Lire_nl (ft, x)

 Ecrire (x)

FinTantque

3) Fin Affiche

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Result ***

1) Spécification du problème :

Résultat = Trait

Traitements =

- ✓ Trait = [nb ← 0]
- a) Afficher (X. Nom, X.Pren) si X.D = 1
- b) Ecrire_nl (ft, X. Nom+ " " + X.Pren) si X.D = 2
- c) nb ← nb + 1 si X.D = 3
- d) X = Lire (f, X)
- ↪ Répéter les tâches a, b, c et d tant qu'il y a des éléments dans le fichier f
- ↪ Afficher (nb)
- ✓ ft = Recréer (ft)
- ✓ f = Ouvrir (f)
- ✓ f et ft = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
nb	Entier	Compteur.
X	Cand	Variable intermédiaire.

2) Algorithme :

0) Procédure Result (Var f : Fich, Var ft : Texte)

1) Ouvrir (f)

2) Recréer (ft)

3) nb ← 0

 Ecrire ("La liste des candidats admis : ")

 Tantque Non(FF(f)) Faire

 Lire (f, X)

 Avec X Faire

 Si D = 1

 Alors

 Ecrire (Nom, " ", Pren)

 Sinon

```

    Si D = 2
    Alors Ecrire_nl (ft, Nom + " " + Pren)
    Sinon nb ← nb + 1
    FinSi
  FinSi
FinAvec
FinTantque

Ecrire ("Le nombre de candidats refusés = ", nb)
4) Fin Result

```

Exercice n°12:

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Program Phrases;

Uses wincrt;

Var

f : Text;

Procedure Saisie (Var x : String);

Begin

Repeat

Write ('Saisir une phrase : ');

Readln(x);

Until (Uppcase(x[1]) In ['A'..'Z']) AND (x[Length(x)] = '.');

End;

Saisie d'une chaîne non vide, commençant par une lettre en majuscule et se terminant par un point.

Procedure Creation (Var f : Text);

Var

x : String;

n, i : Integer;

Begin

Rewrite (f);

Repeat

Write ('Combien de phrases voulez-vous entrer? ');

Readln (n);

Until n>=0;

Writeln ('Une phrase commence par une lettre
et se termine par un "." ');

For i:= 1 to n do

Begin

Writeln ('Phrase n° ',i,' : ');

```
Saisie (x);
Writeln(f, x);
end;
```

End;

```
Procedure Supp (Var x : String);
Begin
  While POS (' ', x) > 0 Do
    Delete (x, Pos (' ', x), 1);
```

End;

```
Procedure Transfert (Var ft, f : Text);
```

```
Var
  x : String;
Begin
  Reset (ft);
  Rewrite (f);

  While Not(EOF(ft)) do
    Begin
      Readln (ft, x);
      Writeln (f, x);
    end;
```

End;

```
Procedure Espaces (Var f : Text);
```

```
Var
  ft : Text;
  x : String;
Begin
  Assign (ft, 'c:\Inter.txt');
  Rewrite (ft);
  Reset (f);

  While Not(EOF(f)) do
    Begin
      Readln (f, x);
      Supp (x);
      Writeln (ft, x);
    end;
```

```
Transfert (ft, f);
```

End;

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Rechercher la position de deux espaces et effacer l'un d'eux. Refaire ce traitement tant qu'il y a deux espaces adjacents dans x.

Avec chaque variable intermédiaire (x), représentant une ligne du fichier ft, la procédure **Supp** permet de supprimer les espaces superflus et ne laisser qu'un seul espace entre deux mots. La lecture d'une ligne déplace le pointeur, de ft, vers la ligne suivante, donc la ligne concernée par les modifications est dépassée !!! Le retour vers cette ligne est interdit (pas d'accès direct avec un fichier texte) !!
L'idée, alors, est d'écrire tous les x lus et/ou modifiés sur un 2^{ème} fichier intermédiaire (ft) et de transférer, à la fin des modifications, tout son contenu vers le fichier d'origine (f).

Procedure Affiche (Var f : Text);

Var x : String;

Begin

Reset (f);

While Not(EOF(f)) do

Begin

Readln (f, x);

Writeln (x);

end;

End;

Procedure Longue (Var f : Text);

Var

x1, x2 : String;

i, nb1, nb2 : Integer;

Begin

Reset(f);

x2 := ""; nb2:=0;

While Not(EOF(f)) do

Begin

Readln (f, x1);

nb1:=0;

For i:= 1 to length(x1) do

If x1[i]=' ' Then Inc(nb1);

If nb1>nb2

Then

begin

nb2 := nb1;

x2 := x1;

end;

end;

Writeln;

Writeln ('La chaîne contenant le plus de mots est :');

Write (x2);

End;

Begin

Assign (f, 'C:\Phrases.txt');

Creation (f);

Espaces (f);

Affiche (f);

Longue (f);

End.

Affichage des lignes du fichier f.

La démarche de la recherche de la plus longue phrase se trouvant dans le fichier (f) consiste à :

1- **Parcourir** le fichier f élément par élément,

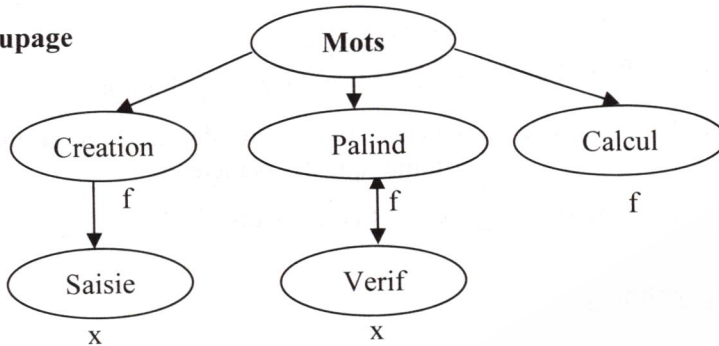
2- **Compter** le nombre d'espaces se trouvant dans chaque ligne,

3- **Comparer** le nombre d'espaces trouvés nb1 avec celui d'une variable nb2 conservant le nombre maximal.

4- **Conserv**er la ligne contenant le maximum d'espaces dans une variable x2.

N.B : Entre deux mots il n'y a qu'un seul espace, donc la phrase qui contient le plus d'espaces c'est la phrase qui contient le plus de mots

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°13:**I- Découpage**

- ✓ **Mots** : programme principal, qui permet d'afficher le nombre de mots, le nombre de palindromes ainsi que la moyenne arithmétique des caractères se trouvant dans un fichier.
- ✓ **Creation** : procédure, permettant de créer un fichier contenant des mots.
- ✓ **Saisie** : procédure, permettant de saisir un mot.
- ✓ **Palind** : procédure, permettant d'afficher le nombre de palindromes se trouvant dans un fichier.
- ✓ **Verif** : fonction, retournant la vérité de la vérification d'un mot, il est palindrome ou non!
- ✓ **Calcul** : procédure, permettant d'afficher le nombre de mots et la moyenne arithmétique des caractères se trouvant dans un fichier.

II- Analyses & Algorithmes

*** Programme principal : Mots ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
 - ↳ Palind (f)
 - ↳ Calcul (f)
- ✓ f = Assigner (f, "C:\Mots.TXT")
- ✓ f = Creation (f)
- ✓ Creation, Palind et Calcul = Modules

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
f	Texte	Contenir des mots.
Palind	Procédure	Afficher le nombre de palindromes.
Calcul	Procédure	Afficher le nombre de mots et la moyenne arithmétique des caractères.
Creation	Procédure	Créer le fichier "Mots.TXT".

2) Algorithme :

- 0) Début Mots
- 1) Assigner (f, " C:\Mots.TXT")
- 2) Creation (f)
- 3) Palind (f)
- 4) Calcul (f)
- 5) Fin Mots

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Creation ***

1) Spécification du problème :

Résultat = f

Traitements =

✓ f = [i ← 0]

↳ Ecrire_nl (f, x) si x ≠ "*" et si i ≤ 100

↳ x = Saisie (x)

↳ i = Compteur, i ↗ +1 à chaque fois où un mot est écrit dans f

↳ Répéter ces dernières tâches jusqu'à x = "*" OU i = 100

✓ f = Recréer (f)

✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
i	Entier	Compteur.
x	Chaîne de caractères	Variable intermédiaire.

2) Algorithme :

0) Procédure Creation (Var f : Texte)

1) Recréer (f)

2) $i \leftarrow 0$

Répéter

Saisie (x)

 Si $x \diamondneq "*" \text{ Alors}$

Ecrire_nl (f, x)

 $i \leftarrow i + 1$

FinSi

Jusqu'à (x = "*") OU (i = 100)

3) Fin Creation

***** Procédure : Saisie *******1) Spécification du problème :****Résultat** = x**Traitements** =✎ x = Donnée avec $x \neq \text{vide}$ et aucun espace dans x,

✓ x = Paramètre formel.

2) Algorithme :

0) Procédure Saisie (VAR X : Chaîne de caractères)

1) Répéter

Ecrire ("Un mot : ")

Lire (x)

 Jusqu'à ($x \diamondneq ""$) ET (POS (" ", x) = 0)

2) Fin Saisie

***** Fonction : Verif *******1) Spécification du problème :****Résultat** = Verif $\leftarrow x1 = x2$ **Traitements** =✓ $x2 = [x2 \leftarrow "" \quad]$ ✎ Affecter les $x1[i]$ dans $x2$ à l'envers✓ i = Compteur allant de 1 à la fin de $x1$ ✓ $x1$ = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
i	Entier	Compteur.
x2	Chaîne de Caractères	Variable intermédiaire.

2) **Algorithme :**

0) Fonction Verif (x1 : Chaîne de caractères) : Booléen

1) $x2 \leftarrow ""$

Pour i de 1 à Long(x1) Faire

 $x2 \leftarrow x1[i] + x2$

FinPour

2) Verif $\leftarrow x1 = x2$

3) Fin Verif

*** Procédure : Palind ***

1) **Spécification du problème :**

Résultat = Afficher (nb)

Traitements =

✓ nb = [nb \leftarrow 0]a) nb $\nearrow +1$ si Verif(x) = Vrai

b) x = Lire_nl (f,x)

↪ Répéter les tâches a et b tant qu'il y a des éléments dans le fichier f

✓ f = Ouvrir (f)

✓ f = Paramètre formel

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
nb	Entier	Compteur.
x	Chaîne de caractères	Variable intermédiaire.

2) **Algorithme :**

0) Procédure Palind (Var f : Texte)

1) Ouvrir (f)

2) nb \leftarrow 0

Tantque Non(FF(f)) Faire

Lire_nl (f, x)

Si Verif(x)

Alors

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

$$nb \leftarrow nb+1$$

FinSi

FinTantque

- 3) Ecrire ("Le nombre de palindromes = ", nb)
- 4) Fin Palind

***** Procédure : Calc *****

1) Spécification du problème :

Résultat = Afficher (nb, s/nb)

Traitements =

✓ s, nb = [s \leftarrow 0 ; nb \leftarrow 0]

a) s \leftarrow s + Long(x) et nb \leftarrow nb + 1

b) x = Lire_nl (f, x)

↪ Répéter les tâches a et b tant qu'il y a des éléments dans le fichier f

✓ f = Ouvrir (f)

✓ f = Paramètre formel

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
s	Entier	Contenir le nombre des caractères.
nb	Entier	Compteur.
x	Chaîne de caractères	Variable intermédiaire.

2) Algorithme :

0) Procédure Calc (Var f : Texte)

1) Ouvrir (f)

2) nb \leftarrow 0

s \leftarrow 0

Tantque Non(FF(f)) Faire

 Lire_nl (f, x)

 s \leftarrow s + Long(x)

 nb \leftarrow nb + 1

FinTantque

3) Ecrire ("Le nombre de mots = ", nb, " et la moyenne arithmétique des caractères = ", s/nb)

4) Fin Calc

3) Traduction en Turbo Pascal**Program** Mots;

Uses WinCRT;

Var

f : Text;

Procedure Saisie (Var x : String);

Begin

Repeat

Write ('Un mot : ');

Readln(x);

Until (x<>'') AND (POS (' ',x)=0);

End;

Saisie d'une chaîne de caractères non vide et ne contenant pas d'espace : un mot.

Procedure Creation (Var f : Text);

Var

i : Integer;

x : String;

Begin

Rewrite(f);

i:=0;

Repeat

Saisie (x);

If x<>'*' then

begin

Writeln (f, x);

Inc (i);

end;

Until (i=100) OR (x='*');

End;

Création d'un fichier texte f, contenant au maximum 100 lignes. L'arrêt du remplissage du fichier est déterminé lorsque le compteur i = 100 ou la chaîne saisie est "*".

www.BAC.org.tn

Page BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

Function Verif (x1 : String): Boolean;

Var

x2 : String;

i : integer;

Begin

x2 := '';

For i := 1 to length(x1) do

x2:=x1[i]+x2;

Verif := x1=x2;**End;**

Verif est une fonction booléenne permettant de vérifier si une chaîne x1 est palindrome ou non.

Procedure Palind (Var f : Text);

Var

nb : Integer;

x : String;

Begin

Reset(f);

nb:= 0;

While Not(EOF(f)) Do

Begin

 Readln (f, x);

 If Verif(x) Then Inc (nb);

end;

Writeln ('Le nombre de palindromes =' ,nb);

End;

Procedure Calc (Var f : Text);

Var

nb, s : Integer;

x : String;

Begin

Reset (f);

nb:=0;

s:=0;

While Not (EOF(f)) Do

Begin

 Readln (f,x);

 Inc (nb);

 s:= s + Length(x);

end;

Writeln ('Le nombre de mots = ' , nb,' et la moyenne arithmétique des caractères = ' , s/nb:2:2);

End;

{Programme principal}

Begin

Assign (f, 'C:\Mots.TXT');

Creation (f);

Calc (f);

Palind (f);

End.

Palind est une procédure permettant de parcourir le fichier f et de compter le nombre de lignes palindromes.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Calc est une procédure permettant d'afficher le nombre de mots (nb) se trouvant dans le fichier f et la moyenne arithmétique des caractères (s/nb) constituant les mots du fichier.

Exercice n°14:**Program** Clients;

Uses wincrt;

Type

Client = Record

Num : Word;

Nom, Pren : String [20];

Sol:real;

End;

Fich = File of Client;

Déclaration d'un nouveau
Type : **Client**.Client est un enregistrement
de 4 champs : Num, Nom,
Pren et Sol.**Var**

f : Fich;

c : Byte;

Procedure Passe;

Begin

Repeat

Gotoxy (30,24);

Write ('Pour continuer tapez Entrer');

Until keypressed;

Readln;

Clrscr;

End;

Passe est une procédure permettant de
donner la main à l'utilisateur de taper la
touche Entrer à fin de passer à un nouvel
écran de traitement.**Clrscr** est une procédure simple
prédéfinie permettant d'effacer le
contenu de la fenêtre d'exécution.**Procedure** Saisie (Var x : Client ; i : Word);

Begin

With x do

Begin

Num := i;

Repeat

Write ('Nom : ');

Readln (Nom);

Until Nom[1] in ['A'..'Z'];

Repeat

Write ('Prénom : ');

Readln (pren);

Until Pren [1] in ['A'..'Z'];

Le champ **Num** est rempli
automatiquement par l'ordinateur
par un entier positif commençant
par 1 et s'incrémentant
automatiquement d'un pas = 1.www.BAC.org.tn

Page BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

```

Write ('Solde : ');
Readln (Sol);
end;
End;
Procedure Creation ( Var f : Fich);

```

www.BAC.org.tn
Page. BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

```

Var
  x : Client;
  i : Word;
  Rep : Char;

```

```
Begin
```

```

{SI-}
Reset(f);

```

```

If IOResult <> 0
Then

```

```
Begin
```

```

Gotoxy (30,2);
Write ('Création du fichier des comptes clients ');
Gotoxy (1,4);
Rewrite (f);
Repeat
  i := Filepos(f)+1;
  Writeln ('Client n ° ',i, ' ===>');
  Saisie (x,i);
  Write (f,x);

```

```
Repeat
```

```

Write ('Autre client (O/N)? : ');
Readln (Rep);

```

```
Until Upcase (Rep) In ['O','N'];
```

```
Until Upcase (Rep) ='N';
```

```

Passe;
end;

```

```

{SI+}
End;

```

L'ouverture d'un fichier en mode **Rewrite** engendre l'écrasement de l'ancien contenu du fichier au cas où ce dernier est déjà créé et rempli par des articles. Afin d'éviter ce genre de situations on doit ouvrir le fichier en mode **Reset**, mais au cas où le fichier n'existe pas (n'est pas encore créé) le compilateur bloque l'exécution du programme et affichera un message réclamant l'inexistence du fichier !

La clé **{SI-}** bloque le compilateur et suite au résultat de la fonction **IOResult**, qui retourne 0 au cas où le fichier est existant et 1 au cas contraire, si elle vaut 1 la procédure Rewrite doit être exécutée sinon Reset est la bonne à exécuter. La clé **{SI+}** débloque le compilateur.

La **réécriture** du fichier f ne se réalise que si le fichier est inexistant (le résultat de la fonction **IORESULT** est différent de zéro).

Gotoxy

La procédure prédéfinie Gotoxy(C, L) permettant de pointer le curseur d'affichage sur la position (C,L) de l'écran d'exécution. C est un entier qui représente le numéro de la **colonne** (il varie entre 1 et 80) et L est un entier qui représente le numéro de la **ligne** (il varie entre 1 et 24).

Procedure Ajout (Var f : Fich);

Var

x : Client;

i : word;

Begin

Reset (f);

Gotoxy (30,2);

Write ('Ajout d"un compte');

Seek(f,Filesize(f));

i :=filepos(f)+1;

Gotoxy (1, 4);

Writeln ('Client n ° ,i, ' ===>');

Saisie (x, i);

Write (f, x);

Passe;

End;

Procedure Transfert (Var ft,f :Fich);

Var

x : Client;

Begin

Reset (ft);

Rewrite (f);

While Not(EOF(ft)) do

Begin

Read (ft, x);

Write (f,x);

end;

End;

www.BAC.org.tn

Page BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

Pointer à la fin du fichier f pour ajouter un client (à la fin).
i est le numéro du nouveau client.

Pour copier le fichier ft dans le fichier f, le fichier ft est ouvert en mode **existant** par contre le fichier f en mode **nouveau**.

C'est un transfert du contenu de ft dans f : l'élément (x) lu à partir du fichier ft est écrit dans le fichier f.

Procédure Supp (Var f:Fich);

Var

i, n : Word;

x : Client;

ft : Fich;

Rep : Char;

Begin

Assign (ft,'C:\Inter.DAT');

Rewrite (ft);

Reset (f);

If EOF(f) Then

Begin

Gotoxy (50,18);

Write ('Fichier vide !!!');

end

else

Begin

Gotoxy (30,2);

Write ('Suppression d'un compte');

Gotoxy (1,4);

Repeat

Write ('Numéro du client à supprimer ? : ');

Readln (n);

Until (n>0) And (n <= Filesize (f));

Repeat

Write ('Etes-vous sûr de vouloir supprimer ce compte (O/N)? : ');

Readln (Rep);

Until Upcase (Rep) In ['O','N'];

IF Upcase (Rep) = 'O' Then

Begin

While filepos (f) <> n-1 do

Begin

Read (f,x);

Write (ft,x);

end;

Read (f,x);

i := n-1;

While Not (Eof(f)) do

Begin

Inc (i);

La suppression d'un élément du fichier f nécessite l'utilisation d'un fichier intermédiaire et local ft.

Tester si le fichier est vide ou non!

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Copier tous les articles qui précèdent celui à supprimer dans le fichier intermédiaire ft.

Lire l'article à supprimer sans le copier dans ft.


```

Read (f,x);
x.num:=i;
Write (ft,x);
end;
Transfert (ft,f);
end;
Passé;
end;
End;
Procedure Modif ( Var f:Fich);
Var
  x : Client;
  n : Word;
Begin
  Reset (f);
  If EOF(f)
  Then
  Begin
    Gotoxy (50,18);
    Write ('Fichier vide !!!');
  end
  Else
  Begin
    Gotoxy (30,2);
    Write ('Modification du solde d"un compte');

    Gotoxy (1,4);
    Repeat
      Write ('Numéro du compte à modifier ? : ');
      Readln (n);
    Until (n>0) And (n <= Filesize (f));

    Seek (f,n-1);
    Read (f,x);
    Write ('Nouveau Solde : ');
    Readln (x.Sol);

    Seek (f, filepos(f)-1);
    Write (f,x);
  Passé;
end;
End;

```

Copier la 2^{ème} tranche du fichier f (les articles qui succèdent l'article à supprimer) dans le fichier ft, avec la mise à jour des numéros des clients de cette tranche.

Transférer tous les articles de ft vers le fichier parent f.

Tester si le fichier est vide ou non!

Pointer et lire l'article à modifier.

La lecture de l'article à modifier déplace le pointeur vers l'article suivant, pour cette raison il faut le reculer d'un article en arrière pour revenir à l'article concerné par la modification.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Procedure Aff (x : Client);

Begin

With x do

Begin

Writeln ('Num = ', Num);

Writeln ('Nom = ', Nom);

Writeln ('Prénom = ', Pren);

Writeln ('Solde = ', Sol:2:2);

end;

Writeln;

End;

Procedure Affiche (Var f : Fich);

Var

x : Client;

n : Word;

Begin

Reset (f);

If EOF(f)

Then

Begin

Gotoxy (50,18);

Write ('Fichier vide !!!');

end

Else

begin

Gotoxy (30,2);

Write ('Affichage d"un compte');

Gotoxy (1,4);

Repeat

Write ('Numéro du client à afficher ? : ');

Readln (n);

Until (n>0) And (n<=Filesize (f));

Seek (f,n-1);

Read (f, x);

Aff (x);

Tester si le fichier est vide ou non!

Pointer et lire l'article à afficher.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

```

If x.sol < 0
Then
  Writeln (' **** Débiteur ****')
Else
  Writeln (' **** Crédeur ****');

```

Passé;

end;

End;

Procedure Affiches (Var f : Fich);

Var

x : Client;

Begin

Reset (f);

If EOF(f) Then

Begin

Gotoxy (50,18);

Write ('Fichier vide !!!');

end

Else

begin

Gotoxy (30,2);

Write ('Affichage de tous les comptes');

Gotoxy (1,4);

While Not (EOF(f)) do

Begin

Read (f,x);

Aff (x);

If Filepos (f) MOD 4 =0

Then **Passé;**

end;

Passé;

end;

End;

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Le fichier peut contenir plusieurs clients, afin d'afficher les données de ces derniers d'une façon lisible et gérable par l'utilisateur, l'ordinateur les affiche en séquence de quatre clients au maximum et à l'utilisateur de les faire défiler avec la vitesse qu'il désire.

Procedure Fin;

Begin

Clrscr;

Gotoxy (38,10);

Write ('Essentiel Banque');

Gotoxy (30,12);

Write ('Fin gestion des comptes clients');

Gotoxy (41,14);

Write ('A Bientôt');

End;**Procédure Fin**

C'est une procédure simple (sans paramètres) permettant d'afficher la dernière image écran que le programme affichera lorsque l'utilisateur quittera l'exécution de cette application.

Procedure Choix (Var c : byte);

Begin

Gotoxy (30, 2);

Write ('Menu principal');

Gotoxy (1,4);

Repeat

Writeln ('1- Ajouter un compte ');

Writeln ('2- Consulter un compte ');

Writeln ('3- Modifier le solde d'un compte ');

Writeln ('4- Supprimer un compte ');

Writeln ('5- Afficher tous les comptes ');

Writeln ('6- Quitter ');

Write ('Tapez le numéro du choix : ');

Readln (C);

Until C In [1..6];

Clrscr;

End;**Procédure choix**

Permet d'afficher un menu à choix multiple à l'utilisateur et de lire son choix.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Begin

Assign (f,'C: \Client.DAT');

Creation (f);

Repeat

Choix (c);

Case c of

1: Ajout (f);

2: Affiche (f);

3: Modif (f);

4: Supp (f);

5: Affiches (f);

6: Fin;

end;

Until c =6;

End.

La création du fichier n'est pas laissée au choix de l'utilisateur mais aussi toutes les mesures de contrôle sont prises en considération au sein de la procédure **Creation**.

L'exécution de l'une des tâches de la gestion d'un ou des comptes d'un ou de plusieurs clients de la banque dépendra du choix de l'utilisateur.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502



"N'attribuez jamais à la malveillance ce qui s'explique très bien par l'incompétence. "

Napoléon Bonaparte

Exercices

Exercice n°1 :

Soit l'algorithme de la procédure "Quoi" suivante :

- 0) Procédure Quoi (Var x : Entier)
- 1) Ecrire ("Saisir un entier > 0 : ")
Lire (x)
- 2) Si $x \leq 0$ Alors Quoi (x) FinSi
- 3) Fin quoi

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Question :

1. Quel est le rôle de la procédure Quoi?
2. Traduire la procédure Quoi en Turbo Pascal.

Exercice n°2 :

Soit la procédure "Quoi", écrite en Turbo Pascal, suivante :

```
Procedure Quoi (x, i : Integer; Var V : TAB);  
Begin  
  Write ('Saisir l'entier n° ', i, ' : ');  
  Readln (V[i]);  
  If i < n Then Quoi (x, i+1, V);  
End;
```

Questions :

1. Quel est le rôle de la procédure Quoi?
2. Reformuler cette même procédure sans utiliser la variable i et sans modifier son objectif de traitement.
3. Reformuler cette même procédure afin que les valeurs saisies soient positives et paires.

Exercice n°3 :

Soit la procédure "Quoi", écrite en Turbo Pascal, suivante :

Procédure Quoi (n, m, i, j : integer; V : BAT);

Begin

If Not(Upcase(V[i,j]) in ['A'..'Z'])

Then Write (V[i,j] : 4);

If (j < m) Then **Quoi (n, m,i,j+1, V)**

Else If (i < n) Then **Quoi (n,m,i+1,1, V)**

End;

Questions :

1. Quel est le rôle de la procédure Quoi?
2. Ecrire l'instruction d'appel utilisée dans le programme appelant à la procédure Quoi?

Exercice n°4 :

Soit la procédure "Quoi", écrite en Turbo Pascal, suivante :

Procédure Quoi (Var f : Fich; n: Word);

Var

x : Don;

Begin

Reset (f);

Seek (f,n);

Read (f,x);

With x do

Begin

Write ('Nom = ',Nom,' *** Prénom = ',Pren);

end;

If Not (EOF(f)) Then **Affich(f , Filepos(f));**

End;

Questions :

1. Quel est le rôle de la procédure Quoi?
2. Ecrire l'instruction d'appel utilisée dans le programme appelant à la procédure Quoi?

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°5 :

Un programme **Commence** utilise une procédure récursive **Aff_let** pour afficher les chaînes de caractères, commençant par la lettre "A", se trouvant dans un tableau T de type TAB, rempli par n chaînes de caractères.

Questions :

- Expliquer par un diagramme la relation entre le programme **Commence** et la procédure **Aff_let**.
- Faire l'analyse et déduire l'algorithme de la procédure **Aff_let**.

Exemple : Soit T dont le contenu est comme suit :

Allo	Livre	Ali	Arôme
1	2	3	4

↳ La procédure affichera : **Allo, Ali, Arôme**

Exercice n°6 :

Soit f un fichier contenant les données relatives à des livres d'une bibliothèque (Titre, Auteur, nombre de pages, ...).

Faire l'analyse et déduire l'algorithme d'une procédure récursive permettant de copier tous les champs "Titre" des éléments du fichier f dans un deuxième fichier texte (ft).

Exercice n°7 :

Soit l'algorithme de la fonction "Tache" suivante :

- Fonction Tache (n,i: entier ; T : TAB) : EntierLong
- Si $i > n$

Alors Tache $\leftarrow 0$

Sinon Tache $\leftarrow T[i] + \text{Tache}(n, i+1, T)$

FinSi

- Fin Tache

Questions :

- Quel est le rôle de la fonction Tache?
- Supposons que le contenu du tableau T de 3 entiers est comme suit :

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T	200	6	5
	1	2	3

Dresser la trace de la pile, avec ses deux phases d'empilement et de dépilement, au cours de l'exécution de la fonction Tache.

3. Reformuler cette même fonction sans utiliser la variable i et sans modifier son objectif de traitement.

Exercice n°8 :

Soit la fonction "Tache", écrite en Turbo Pascal, suivante :

Function Tache (n: integer; T: TAB) : integer;

Begin

If n=0 Then Tache := 0

Else If (T[i] Mod 3 = 0) Then Tache := 1+Tache(n-1,T)

Else Tache := Tache(n-1,T);

End;

Questions :

1. Quel est le rôle de la fonction Tache?
2. Supposons que le contenu du tableau T de 4 entiers soit comme suit :

T	12	40	15	6
	1	2	3	4

Dresser la trace de la pile, avec ses deux phases d'empilement et de dépilement, au cours de l'exécution de la fonction Tache.

Exercice n°9 :

Soit l'algorithme de la fonction "Tache" suivante :

0) Fonction Tache (C: Chaîne) : Chaîne

1) P ← Pos ("^^",C)

Si P=0

Alors Tache ← C

Sinon

Efface(C,p,1)

Tache ← C

FinSi

2) Fin Tache

N.B : Le symbole "^^" représente le caractère espace.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature / Type	Rôle
P	Entier	Contenir

Questions :

1. Quel est le rôle de la fonction Tache?
2. Supposons que le contenu initial de la chaîne C soit :

"4^^^Sciences^^de^l'informatique"

Quel est le résultat final retourné par cette fonction?

3. Faire la traduction de la fonction Tache en Turbo pascal.

Exercice n°10 :

Soit l'algorithme de la fonction "Tache" suivante :

0) Fonction Tache (n, x : Entier ; T: Tab) : Booléen

1) Si n = 0

Alors

Tache ← Faux

Sinon Si T[n] = x

Alors

Tache ← Vrai

Sinon

Tache ← Tache (n-1, x, T)

FinSi

2) Fin Tache

Questions :

1. Quel est le rôle de la fonction Tache?
2. Faire la traduction de la fonction Tache en Turbo pascal

Exercice n°11 :

Un programme **Nombre** utilise une fonction **récursive Nb_2Chiff** pour calculer le nombre d'entiers de deux chiffres, se trouvant dans un tableau T de type TAB contenant n entiers.

- a) Expliquer par un diagramme la relation entre le programme **Nombre** et la fonction **Nb_2Chiff**.
- b) Faire l'analyse et déduire l'algorithme de la fonction **Nb_2Chiff**.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°12 :

Faire l'analyse, déduire les algorithmes et faire la traduction, en Turbo Pascal, d'un programme permettant d'afficher sous forme d'un triangle rectangle une chaîne de caractères (ch) non vide et de 15 caractères **numériques** au maximum.

Exemple :

Si ch = "2546785"

L'ordinateur affichera :

```
2546785
254678
25467
2546
254
25
2
```

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

N.B : La solution ne doit pas comporter des structures itératives.

Exercice n°13 :

Ecrire un programme, en Turbo Pascal, permettant de :

- ✓ Remplir un fichier texte, par des phrases non vides (maximum 50), dans la racine "C:" sous le nom de "Ph_chiff.TXT"
- ✓ Affecter le nombre de chiffres se trouvant dans chaque ligne du fichier dans un tableau T,
- ✓ Afficher le nombre de chiffres utilisés dans les lignes du fichier.

Exemple :

Supposons que le fichier contienne les 5 lignes suivantes :

- ↺ Clavier
- ↺ 10 Août 2004
- ↺ 4^{ème} année Sciences de l'informatique.
- ↺ Bac SI 2010
- ↺ 1 ver vert dans 1 verre vert.

Le contenu de T est :

0	6	1	4	2
---	---	---	---	---

L'ordinateur affichera le nombre 13.

N.B : La solution ne doit pas comporter des structures itératives.

Corrigés

Exercice n°1 :

1. La procédure "Quoi" permet de remplir la variable x, à partir du clavier, par un entier naturel et non nul.

Explications : Le traitement élémentaire de la procédure est la lecture d'une valeur strictement positive, au cas où la valeur saisie est négative ou nulle ($x \leq 0$), la procédure "Quoi" sera exécutée de nouveau pour le même objectif. Parcontre, si la valeur saisie est bonne le problème est résolu et ce sera la fin de l'opération.

2. Traduction de la procédure "Quoi" en Turbo Pascal.

Procedure Quoi (Var x : integer);

Begin

Write ('Saisir un entier > 0 : ');

Readln (x);

IF $x \leq 0$ Then Quoi (x);

End;

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°2 :

1. La procédure "Quoi" permet de remplir un tableau V, à une dimension, par n entiers relatifs ($\in \mathbb{Z}$).

Explications : Le traitement élémentaire de la procédure est le remplissage d'une case $V[i]$ par un entier relatif. Cette tâche sera répétée n fois en exécutant de nouveau la procédure "Quoi" tout en incrémentant le compteur i par 1.

2. La procédure "Quoi" sans utiliser la variable i :

Procedure Quoi (x : Integer; Var V : TAB);

Begin

Write ('Saisir l'entier n° ', x, ' : ');

Readln (V[x]);

If $x > 1$ Then Quoi (x-1, V);

End;

Explications : Le tableau V sera rempli de la case n° x jusqu'à la case n°1, autrement dit remplissage "en marche arrière!". A chaque fois où une case du tableau V est remplie, l'ordinateur refait la même tâche mais en passant à une nouvelle case du tableau V, celle dont le numéro = x-1 et ainsi de suite jusqu'à x = 1, c'est la fin des traitements.

3. Remplissage de V par des entiers positifs et pairs.

Procedure Quoi (x : Integer; Var V : TAB);

Begin

Write ('Saisir l'entier n° ',x,' : ');

Readln (V[x]);

If (Not (V[x] > 0)) AND (Not (V[x] MOD 2 = 0))

Then **Quoi** (x, V)

Else If x > 1

Then **Quoi** (x-1,V);

End;

Explications : A chaque fois où une case du tableau V est remplie, l'ordinateur teste la valeur saisie et se trouve devant trois eventuelles situations :

- ↪ Refaire le même traitement avec la même case : la valeur, récemment saisie dans cette case, n'est pas valide (**Quoi(x,V)**),
- ↪ Continuer les traitements avec une nouvelle case du tableau V (**Quoi(x-1,V)**), celle dont le numéro = (x-1) si x > 1,
- ↪ Arrêter les traitements, c'est la **fin** de la procédure Quoi.

Exercice n°3 :

1. La procédure Quoi permet d'afficher tous les **caractères non alphabétiques** se trouvant dans une matrice V.
2. L'instruction d'appel utilisée dans le programme appelant à la procédure Quoi : **Quoi (n, m, 1, 1, V)**;

Explications : Dans l'instruction d'appel à la procédure Quoi cinq paramètres effectifs sont utilisés :

n: Le nombre de lignes

m: Le nombre de colonnes

- 1 : La valeur initiale du compteur i
- 1 : La valeur initiale du compteur j
- V : La matrice contenant nXm caractères

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°4 :

1. La procédure **Quoi** permet d'afficher les champs nom et prénom de tous les articles se trouvant dans le fichier f.
2. L'instruction d'appel utilisée dans le programme appelant à la procédure **Quoi** : **Quoi (f, 0)**;

Explications : Dans l'instruction d'appel à la procédure **Quoi** deux paramètres effectifs sont utilisés :

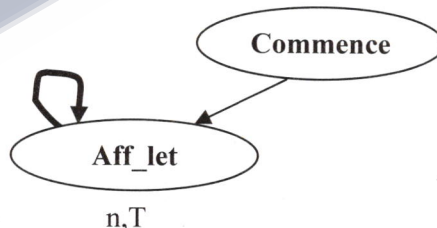
f: Le fichier contenant des données

0: (zéro) c'est le numéro du premier article du fichier f.

L'instruction **Seek(f, n)** est utilisée pour pointer sur l'article à traiter (numéro n). Initialement, c'est l'article numéro 0 et au cours des traitements ce numéro sera déterminé via la fonction prédéfinie **Filepos(f)** qui retourne la position actuelle du pointeur sur le fichier.

Exercice n°5 :

- a) Le diagramme expliquant la relation entre le programme **Commence** et la procédure **Aff_let** :
- b)



La première flèche unidirectionnelle passant du programme appelant "Commence" vers le module "Aff_let" indique qu'il s'agit d'un appel à une procédure et la deuxième flèche se trouvant sur l'entité de la procédure **Aff_let** indique qu'il s'agit d'une procédure récursive.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

c) Analyse et algorithme de la procédure **Aff_let** :

1) **Spécification du problème :**

Résultat = Afficher toutes les chaînes se trouvant dans T et commençant par "A"

Traitements =

- ✓ Ecrire (T[i]) Si T[i][1]='A'
- ✓ i = Compteur allant de 1 à n
- ✓ n, i et T = Paramètres formels

2) **Algorithme :**

0) Procédure **Aff_let** (n, i : entier ; T : TAB)

1) Si T[i][1] = "A"

Alors

Ecrire (T[i])

FinSi

2) Si $i < n$ Alors **Aff_let** (n, i+1, T) Finsi

3) Fin **Aff_let**

Exercice n°6 :

1) **Spécification du problème :**

Résultat = ft

Traitements =

- ✓ ft = []
- ↪ Ecrire_nl (ft, x.Titre)
- ↪ Ajouter (ft) : Ouvrir le fichier (ft) en mode ajout
- ✓ x = Lire (f, x) c'est l'élément pointé de numéro n
- ↪ n = renouveler par un paramètre effectif déterminé à travers la fonction prédéfinie Position_fichier (f)
- ✓ f = Ouvrir (f) : Ouvrir le fichier (f) en mode existant
- ↪ A chaque fois un élément du fichier est pointé = Rechercher_position (f, n)
- ✓ n, f et ft = Paramètres formels

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

2) **Algorithme :**0) Procédure Transfert (n : Mot ; VAR ft : Texte ; VAR f : Fich)1) Ajouter (ft)2) Ouvrir (f)3) Rechercher_position (f, n)Lire (f, x)Ecrire_nl (ft, x, Titre)4) Si Non (FF(f))Alors Transfert (Position_fichier (f, ft, f))

FinSi

5) Fin Transfert

Actualiser la valeur de n par la position actuelle du pointeur, c'est le dernier élément pointé et qui servira pour la continuité du traitement.

Exercice n°7 :www.BAC.org.tn

Page BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

1. La fonction Tache permet de retourner au programme appelant la somme des éléments se trouvant dans le tableau unidimensionnel T.

2. Soit le contenu du tableau T de 3 entiers comme suit :

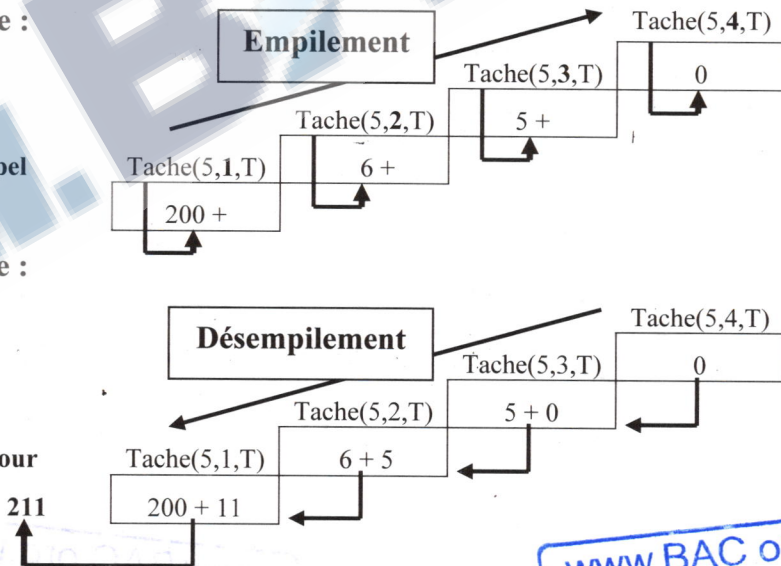
T	200	6	5
	1	2	3

Pile :

Appel

Pile :

Retour



www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

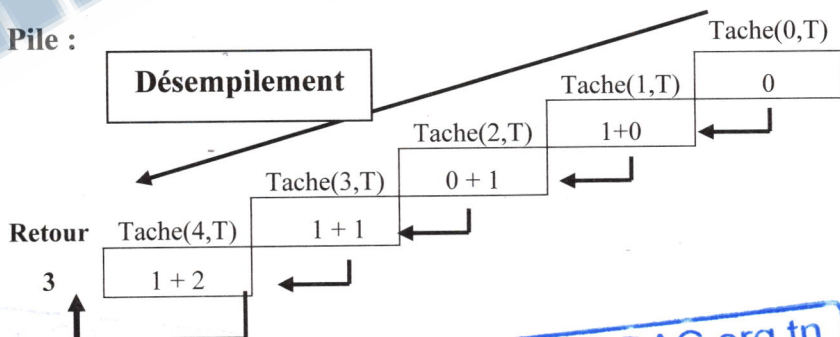
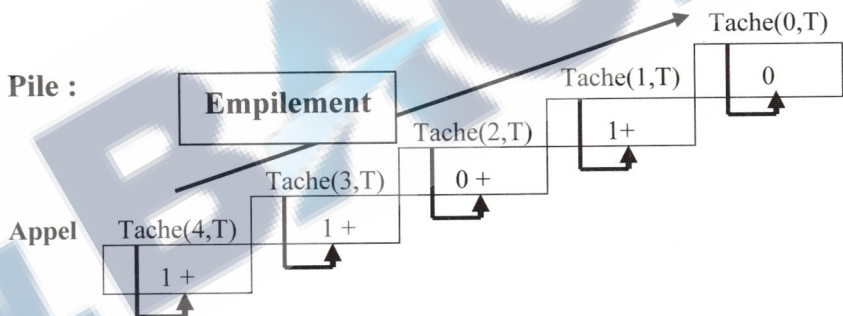
3. La fonction Tache à l'exclusion de la variable i :
- 0) Fonction Tache (n: entier ; T : TAB) : EntierLong
- 1) Si n= 0
 Alors Tache \leftarrow 0
 Sinon Tache \leftarrow T[n] + Tache(n-1,T)
 FinSi
- 2) Fin Tache

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Exercice n°8 :

1. La fonction Tache permet de retourner au programme appelant le **nombre d'éléments**, se trouvant dans le tableau unidimensionnel T, qui sont **multiples de trois**.
2. Soit le contenu du tableau T de 4 entiers comme suit :

T	12	40	15	6
	1	2	3	4



www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Exercice n°9 :

1. La fonction Tache permet de retourner au programme appelant **une chaîne de caractères ôtée des espaces superflus (inutiles)**.
2. Soit le contenu initial de la chaîne C comme suit :

"4^^^Sciences^^de^l'informatique"

Le résultat final retourné par cette fonction est :

"4^Sciences^de^l'informatique"

3. La fonction Tache en Turbo Pascal :

Function Tache (C: String) : String;

Var

 P : integer;

Begin

P := Pos (' ', C);

If P=0

 Then

 Tache := C

 Else

 Begin

 Delete(C, p, 1);

 Tache := C;

 end;

End;

Exercice n°10 :

1. La fonction Tache permet de retourner au programme appelant la vérité de l'existence d'un entier x dans le tableau T de n entiers.
2. La fonction Tache en Turbo Pascal :

Function Tache (n, x : integer ; T: Tab) : Boolean;

Begin

If n = 0

Then

Tache := False

Else if T[n] = x

Then

Tache := True

Else

Tache := Tache (n-1, x, T);

End;

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°11 :

- a) Le diagramme exprimant la relation entre le programme Nombre et la fonction récursive Nb_2Chiff est le suivant :



- b) Analyse & Algorithme de la fonction récursive Nb_2Chiff :

1) **Spécification du problème :**

Résultat = Nb_2Chiff

Traitements =

- ✓ **Nb_2Chiff** = []
- ↳ $Nb_2Chiff \leftarrow 0$ si $n = 0$,
- ↳ $Nb_2Chiff \leftarrow 1 + Nb_2Chiff(n-1, T)$ Si T[n] est de deux chiffres,
- ↳ $Nb_2Chiff \leftarrow 0 + Nb_2Chiff(n-1, T)$ Sinon,
- ✓ n et T = Paramètres formels

2) Algorithme :

0) Fonction Nb_2Chiff (n : entier ; T : Tab): Entier

1) Si n = 0

Alors

Nb_2Chiff \leftarrow 0

Sinon

Si T[i] Dans [10..99]

Alors Nb_2Chiff \leftarrow 1 + Nb_2Chiff (n-1,T)Sinon Nb_2Chiff \leftarrow Nb_2Chiff (n-1,T)

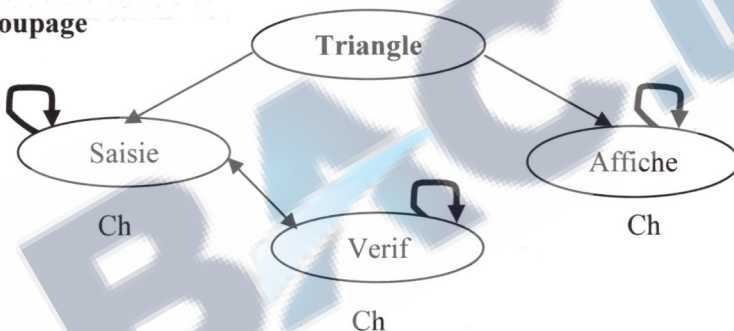
FinSi

FinSi

2) Fin Nb_2Chiff

Exercice n°12 :

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

I- Découpage

- ✓ **Triangle** : programme principal, qui permet d'afficher la chaîne Ch sous la forme d'un triangle rectangle.
- ✓ **Saisie** : procédure récursive, permettant de saisir une chaîne de caractères non vide et de 15 caractères au maximum.
- ✓ **Affiche** : procédure récursive, permettant d'afficher Ch sous la forme demandée.
- ✓ **Verif** : fonction récursive, retournant la vérité de la pureté de la chaîne est-elle pure ou non (pure : ne contient que des chiffres)

II- Analyses & Algorithmes

*** Programme principal : Triangle ***

1) Spécification du problème :

Résultat = Affiche (Ch)

Traitements =

- ✓ Ch= Saisie (Ch)
- ✓ Saisie et Affiche = modules

T.D.O.U.G

Objet	Nature/Type	Rôle
Ch	Chaîne	Contenir une chaîne de caractères.
Affiche	Procédure	Afficher une chaîne de caractères sous la forme d'un triangle rectangle.
Saisie	Procédure	Saisir une chaîne de caractères.

2) Algorithme :

- 0) Début Triangle
- 1) Saisie (Ch)
- 2) Affiche (Ch)
- 3) Fin Triangle

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

***** Procédure : Saisie *******1) Spécification du problème :****Résultat = Ch****Traitements =**

- ✓ Ch = Donnée avec $\text{Long}(\text{Ch}) \in [3..15]$ ET $\text{Verif}(\text{Ch}) = \text{Vrai}$
- ✓ $\text{Verif} = \text{Fonction}$
- ✓ Ch = Paramètre formel

T.D.O.U.L

Objet	Nature/Type	Rôle
Verif	Fonction/Booléen	Retourner la vérité de la pureté de Ch.

2) Algorithme :

- 0) Procédure Saisie (VAR Ch : Chaîne)
 - 1) Ecrire ("Saisir une chaîne de caractères non vide et de 15 chiffres au maximum: ")
Lire (Ch)
 - 2) Si (NON (Long (Ch) Dans [3..15])) OU (NON(Verif(Ch)))
Alors
 Saisie (Ch)
FinSi
 - 3) Fin Saisie

*** Fonction : Verif ***

1) Spécification du problème :

Résultat = Verif

Traitements =

- ✓ Verif = []
 - ↳ Verif ← Vrai si la chaîne est devenue vide,
 - ↳ Verif ← Faux Si Ch[1] n'est pas un chiffre,
 - ↳ Verif ← Verif (Sous_chaine(Ch, 2, Long(ch)-1)) Sinon,
- ✓ Ch = Paramètre formel

2) Algorithme :0) Fonction **Verif** (Ch : chaîne): Booléen

1) Si Ch = ""

Alors

Verif ← Vrai

Sinon

Si NON (Ch[1] Dans ["0".."9"])

Alors

Verif ← Faux

Sinon

Verif ← Verif (Sous_chaine (Ch, 2, Long (ch)-1))

FinSi

FinSi

2) Fin Verif

*** Procédure : Affiche ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
 - ↳ Afficher La chaîne Ch et faire un retour à la ligne, initialement ch est égale à toute la chaîne
 - ↳ Si la chaîne n'est pas vide la procédure s'exécutera de nouveau mais avec une copie de la chaîne = la chaîne Ch dépourvue du dernier caractère
- ✓ Ch = Paramètre formel

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

2) Algorithme :

0) Procédure Affiche (Ch : Chaîne de caractères)

Ecrire (Ch)

Retour_à_la_ligne

Si Ch <> ""

Alors Affiche (Sous_chaîne (Ch, 1, Long (ch)-1))

FinSi

1) Fin Affiche

2) Traduction en Turbo Pascal :**Program** Triangle;

Uses Winrt;

Var Ch : String;

Function Verif (Ch : String): Boolean;

Begin

If Ch=""

Then Verif := True

Else

IF NOT (Ch[1] In ['0'..'9'])

Then Verif := False

Else Verif := Verif (Copy (Ch, 2, Length (ch)-1));

End;

Procedure Saisie (VAR Ch : String);

Begin

 Write ('Saisir une chaîne de caractères non vide et de 15 chiffres au
 maximum: ');

Readln (Ch);

IF (NOT (Length (Ch) In [3..15])) Or (NOT(Verif(Ch)))

Then Saisie (ch);

End;

Procedure Affiche (Ch : String);

Begin

Writeln (Ch);

IF Ch <> ""

Then Affiche (Copy (Ch, 1, Length (ch)-1));

End;

Begin

Saisie (Ch);

Affiche (Ch);

End.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°13 :**Program** Non_Accent;

Uses wincrt;

Type

TAB = Array [1..50] of integer;

Var

ft: **Text**;

T: TAB;

n: integer;

Procedure Choix (Var c:Char);

Begin

Write('Autres chaînes de caractères (O/N)? : ');

Readln(c);

if Not(Uppcase(c) In ['O','N']) Then **Choix**(c);**End**;**Procedure** Saisie (Var x: String);

Begin

Write ('Saisir une chaîne de caractères non vide : ');

Readln(x);

If x = "" Then **Saisie**(x);**End**;**Function** Nb_Chiff (x : String): integer;

Begin

If x=""

Then

Nb_Chiff:=0

Else

If x[1] in ['0'..'9']

Then Nb_Chiff:=1+Nb_Chiff (copy(x, 2, Length(x)-1))

Else Nb_Chiff:=Nb_Chiff (copy(x, 2, Length(x)-1));

End;**Procedure** Remp (Var ft: Text; Var n: integer ;VAR T : TAB);

Var

x : string;

c: char;

Begin

Append (ft);

La fonction Nb_Chiff permet de compter le nombre de chiffres dans une chaîne de caractères.

Le fichier ft doit être ouvert en mode ajout afin d'y ajouter des lignes à la fin.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

```
Saisie(x);
n:=n+1;
T[n] := Nb_chiff(x);
Writeln (ft, x);
```

```
If n<50
Then
  Begin
    Choix(c);
    If Uppcase(c)='O'
    Then Remp (ft, n, T);
  end;
End;
```

Function Som (n: integer ; T: TAB): integer;

```
Begin
  If n=0
  Then
    Som :=0
  Else
    Som := T[n]+ Som (n-1,T);
End;
```

{Programme Principal}

Begin

```
Assign (ft, 'C:\Ph_chiff.TXT');
Rewrite (ft);
Remp (ft, n, T);
Write ('Le nombre de chiffres utilisés dans le fichier = ', Som (n, T));
```

End.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

On ne demande l'avis de l'utilisateur, de continuer ou non la saisie, que si $n < 50$.

La fonction **Som** permet de retourner la somme des entiers se trouvant dans T.



"أَتَمَلُّ الدَّفْسَ بِالْأَمَالِ أَرْتَمِبَهَا مَا أَضِيقَ الْعَيْشَ لَوْلَا فَسْعَةُ الْأَمَلِ"
أَلْطَغْرَائِي

Exercices

Exercice n° 1 :

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Faire l'analyse, déduire l'algorithme et faire la traduction, en Turbo Pascal, d'un module **récur­sif** permettant de tester si les n entiers d'un tableau T sont triés dans l'ordre croissant ou non.

N.B : n est un entier et T est de type TAB, des paramètres formels.

Exercice n° 2 :

Ecrire un programme, en Turbo Pascal, permettant de trier dans l'ordre croissant une liste de n entiers positifs et non nuls (avec $n \in [5..50]$). La solution à proposer doit utiliser la méthode de **tri par sélection** et tous les modules utilisés doivent être **récur­sifs**.

Exercice n° 3 :

Faire l'analyse, déduire les algorithmes et faire la traduction, en Turbo Pascal, d'un programme permettant de trier dans l'ordre décroissant une liste de n caractères imprimables (leurs code ASCII $\in [33..127]$ et $n \in [2..60]$). La solution à proposer doit utiliser la méthode de **tri à bulles** et tous les modules utilisés doivent être **récur­sifs**.

Exercice n° 4 :

Faire l'analyse et déduire les algorithmes d'un programme permettant de trier dans l'ordre alphabétique croissant une liste de n ($n \in [4.. 20]$) chaînes de caractères, chacune de taille $\in [2.. 10]$. La solution à proposer doit utiliser la méthode de **tri par insertion**.

Exercice n° 5 :

Refaire la résolution du problème de l'exercice précédent en créant un programme en Turbo Pascal n'utilisant que des modules **récur­sifs**.

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n° 6 :

Soient T1 et T2 deux tableaux contenant respectivement n et m entiers triés dans l'ordre décroissant avec n et m $\in [3..30]$.

Le tri d'un tableau est réalisé au fur et à mesure de la saisie de ses éléments, un entier n'est accepté que s'il est inférieur à l'entier précédent.

Faire l'analyse et déduire l'algorithme d'un programme qui permet de **fusionner** le contenu des deux tableaux dans un troisième tableau T, de telle manière que le tableau T soit trié dans l'ordre décroissant.

N.B : Il faut choisir la **méthode de Tri par fusion** qui garantit l'obtention d'un tableau T trié.

Exemple : Supposons que T1 et T2 soient comme suit:

T1	87	40	0
	1	2	3

T2	156	100	1	-7	-8
	1	2	3	4	5

Suite à la fusion de T1 et T2 le tableau T sera comme suit :

T	156	100	87	40	1	0	-7	-8
	1	2	3	4	5	6	7	8

Exercice n° 7 :

Soit T, un tableau de n réels aléatoires (choisis par l'ordinateur et qui soient entre 0 et 1) avec n $\in [10..100]$. Ecrire un programme, en Turbo Pascal, qui permet de trier dans l'ordre croissant le tableau T. La solution à proposer doit utiliser la méthode de tri **SHELL**.

Exercice n° 8 :

Soit un vecteur V de n entiers pairs, avec $n \in [10..2000]$. Faire l'analyse et déduire les algorithmes d'un programme qui permet de vérifier si une valeur x , paire et saisie au clavier, existe dans le vecteur V ou non.

La solution à proposer doit utiliser la méthode de **recherche dichotomique (récursive)**.

Exercice n° 9 :

Soit le fichier "Eleves.FCH", enregistré dans la partition "C:", contenant les données relatives aux élèves d'une classe.

Chaque élève est caractérisé par :

- ↪ Un **nom** : chaîne de 20 cc au maximum,
- ↪ Un **prénom** : chaîne de 20 cc au maximum,
- ↪ Une **moyenne** : Réel $\in [0..20]$.

Ecrire un programme en Turbo Pascal permettant de trier tous les élèves dans l'ordre alphabétique croissant et de les afficher. Le critère de tri est le nom puis le prénom de l'élève.

Exemple : L'exécution de ce programme donne :

N°	Nom	Prénom	Moyenne
1	Bahri	Marwa	11.25
2	Ben Ismaïl	Ghada	12.58
3	Ben Ismaïl	Ghanem	14.75
4	Ben Ismaïl	Mongi	13.97
5	Boukadida	Jihène	15.20
6	Harbaoui	Fathia	14.75
7	Jawadi	Bessem	16.70
8	Kaddachi	Heifa	16.35
9	Rahmouni	Kaïs	12.30

...

Exercice n° 10:

Faire l'analyse et déduire les algorithmes d'un programme qui crée sur le disque dur (partition "C:") un fichier dont le nom physique est "Stock.FCH".

Le fichier "Stock.FCH" contient les données des accessoires d'automobiles, se trouvant dans un stock d'un revendeur et qui sont **triés au fur et à mesure de leur saisie**, dans l'ordre croissant de leurs désignations.

Un accessoire d'une automobile est caractérisé par :

- Un **numéro d'ordre** : Entier positif, attribué par l'ordinateur,
- Une **désignation** : Chaîne de 25 cc au maximum,
- Un **prix unitaire** : Réel positif.

Le programme doit assurer, en plus, les tâches de gestion suivantes :

- ✓ L'affichage de tous les articles du fichier,
- ✓ L'affichage des articles dont la désignation commence par un caractère donné,
- ✓ L'affichage des articles dont le prix unitaire se trouve dans une marge de prix donnée.

N.B :

Lors de la création, l'ajout d'un article à une position du fichier nécessite la modification des numéros d'ordre de tous les articles successeurs.

www.BAC.org.tn

Page: BAC-TUNISIE

Tél: 25 361 197 / 53 371 502

www.BAC.org.tn
BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Corrigés

Exercice n°1 :

Analyse & Algorithme de la fonction récursive **Verif** permettant de vérifier si les n entiers du tableau T sont triés dans l'ordre croissant ou non :

1) Spécification du problème :

Résultat = Verif

Traitements =

- ✓ **Verif** = []
- ↗ Verif \leftarrow Vrai si $n = 1$,
- ↗ Verif \leftarrow Faux Si $n \neq 1$ et Si $T[n] < T[n-1]$,
- ↗ Verif \leftarrow Verif ($n-1, T$) Si $n \neq 1$ et Si $T[n] \geq T[n-1]$,
- ✓ n et T = Paramètres formels

2) Algorithme :

0) Fonction Verif (n : entier ; T : Tab): Booléen

- 1) Si $n = 1$
 - Alors
 - Verif \leftarrow Vrai
 - Sinon
 - Si $T[n] < T[n-1]$
 - Alors
 - Verif \leftarrow Faux
 - Sinon
 - Verif \leftarrow Verif ($n-1, T$)
 - FinSi
 - FinSi
- 2) Fin Verif

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

3) Traduction en Turbo Pascal :**Function** Verif (n : integer ; T : Tab): Boolean;

Begin

```

    If n = 1
    Then
        Verif := True
    Else
        If T[n] < T[n-1]
        Then
            Verif := False
        Else
            Verif := Verif (n-1, T);

```

End;**Exercice n°2 :****Program** Tri_selec;

Uses wincrt;

Type

Tab = Array [1..50] of integer;

Var

n, d : integer;

T: Tab;

Procedure Saisie (Var n : integer);

Begin

Writeln ('Saisir un entiere entre 5 et 50 :');

Readln (n);

If **Not** (n In [5..50]) Then **Saisie** (n);**End;****Procedure** Remplir (Var T : Tab ; n : integer);

Begin

Writeln ('T['n,']=');

Readln (T[n]);

If T[n] <= 0

Then

Remplir (T, n)

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502


```
Else
  If n > 1
    Then
      Remplir (T, n-1);
```

```
End;
```

```
Procedure Minimum (T : Tab; n, d : integer; Var min :integer);
```

```
Begin
```

```
  If T[d] < T[min]
```

```
  Then
```

```
    min:= d;
```

```
  If d < n
```

```
  Then
```

```
    Minimum (T , n, d+1 , min );
```

```
End;
```

```
Procedure Permut (Var a, b: integer);
```

```
Var
```

```
  c : integer ;
```

```
Begin
```

```
  c:=a;
```

```
  a:=b;
```

```
  b:=c;
```

```
End;
```

```
Procedure Tri (Var T:Tab ; d,n : integer);
```

```
Var
```

```
  min: integer;
```

```
Begin
```

```
  min := d;
```

```
  Minimum (T, n, d+1, min);
```

```
  If (min <> d)
```

```
  Then
```

```
    Permut (T[min],T[d]);
```

```
  If (d < n-1)
```

```
  Then
```

```
    Tri (T, d+1, n);
```

```
End;
```

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

```

Procedure Affiche (T: Tab ; n, i : integer);
Begin
  Write (T[i]:5);
  If i<n
  Then Affiche (T, n, i+1);
End;

```

{Programme principal}

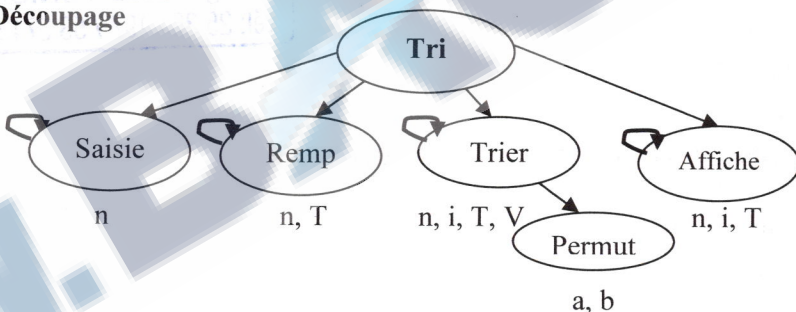
```

Begin
  Saisie (n);
  Remplir (T, n);
  Tri (T, 1, n);
  Clrscr;
  Writeln ('*** Le contenu de T trié dans l'ordre croissant : *** ');
  Affiche (T, n, 1);
End.

```

Exercice n°3 :

I- Découpage



- ✓ **Tri** : Programme principal, qui permet d'afficher le contenu de T trié.
- ✓ **Saisie** : procédure récursive, permettant de saisir la valeur de n.
- ✓ **Remp** : procédure récursive, permettant de remplir le tableau T par n caractères imprimables.
- ✓ **Trier** : procédure récursive, permettant de trier le tableau T dans l'ordre décroissant.
- ✓ **Affiche** : procédure récursive, permettant d'afficher le contenu du tableau T.
- ✓ **Permut** : procédure, permettant de permuter deux caractères a et b.

II- Analyses & Algorithmes

*** Programme principal : Tri ***

1) Spécification du problème :

Résultat = Affiche (T, n, 1) : Le compteur i est initialisé à 1.

Traitements =

- ✓ n = Saisie (n)
- ✓ T = Remp (n, T)
- ✓ T = Trier (n, 1, T, Vrai)
- ✎ Le compteur i est initialisé à 1 et la variable v est initialisée à vrai
- ✓ Affiche, Saisie, Remp et Trier = Procédures

T.D.O.U.G

Objet	Nature/Type	Rôle
n	Entier	Contenir le nombre de caractères à trier.
T	TAB	Contenir n caractères imprimables.
Saisie	Procédure	Saisir la valeur de n.
Remp	Procédure	Remplir le tableau T.
Trier	Procédure	Trier le tableau T dans l'ordre décroissant.
Affiche	Procédure	Afficher le contenu du tableau T.

T.D.N.T

Type
TAB = Tableau de 60 caractères

2) Algorithme :

- 0) Début Tri
- 1) Saisie (n)
- 2) Remp (n, T)
- 3) Trier (n, 1, T, Vrai)
- 4) Affiche (n, 1, T)
- 5) Fin Tri

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Saisie ***

1) Spécification du problème :

Résultat = n

Traitements =

- ✓ n = Donnée
- ✓ Au cas où $n \notin [2..60]$ Saisie (n) est exécutée de nouveau!
- ✓ n = Paramètre formel

2) Algorithme :

- 0) Procédure Saisie (Var n : Entier)
- 1) Ecrire ("Saisir le nombre de caractères (entre 2 et 60) : ")
Lire (n)
- 2) Si **NON** (n Dans [2..60])
Alors Saisie (n)
FinSi
- 3) Fin Saisie

*** Procédure : Remp ***

1) Spécification du problème :

Résultat = T[n]

Traitements =

- ✓ T[n] = Donnée

A un moment donné le résultat de la procédure Remp est le remplissage d'un n^{ième} case de T.

- ↪ Si T[n] n'est pas un caractère alors Remp est exécutée de nouveau avec la même case T[n]
- ↪ Si $n > 1$ alors Remp est exécutée de nouveau avec une nouvelle case T[n-1]

- ✓ n et T = Paramètres formels

2) Algorithme :

- 0) Procédure Remp (n : Entier ; VAR T : TAB)
- 1) Ecrire ("Saisir le caractère n^o", n, " : ")
Lire (T[n])
- 2) Si **NON** (ORD (T[i]) Dans [33..127])
Alors **Remp (n, T)**
Sinon
Si $n > 1$
Alors **Remp (n-1, T)**
FinSi
FinSi
- 3) Fin Remp

www.BAC.org.tn
Page. BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Trier ***

1) Spécification du problème :

Résultat = T

Traitements =

✓ T = []

↪ Permuter deux cases consécutives en utilisant Permut (T[i], T[i+1]) si et seulement si T[i] < T[i+1]

↪ Signaler qu'il s'agit de une modification, de l'ordre des éléments de T, en changeant la vérité de la variable booléenne V (V ← Faux).

↪ Si i < n-1 il faut exécuter la procédure Trier de nouveau en incrémentant le compteur i par 1 et en gardant la même valeur de V.

↪ Si i = n-1, c'est la fin d'un parcours de T, si la variable booléenne V = Faux, il faut exécuter de nouveau la procédure Trier mais avec les valeurs initiales de i et de v respectivement 1 et vrai pour réaliser un nouveau parcours de T.

✓ Permut = Procédure

✓ n, i, T et V = Paramètres formels

2) Algorithme :

0) Procédure Trier (n, i : Entier; VAR T: TAB ; V : Booléen)

1) Si T[i] < T[i+1]

Alors

Permut (T[i], T[i+1])

V ← Faux

FinSi

2) Si i < n-1

Alors Trier (n, i+1, T, V)

Sinon

Si V = Faux

Alors Trier (n, 1, T, Vrai)

FinSi

FinSi

3) Fin Trier

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Permut ***

1) Spécification du problème :

Résultat = a, b

Traitements =

✓ a, b = []

↵ $a \leftarrow b$

↵ $b \leftarrow c$

↵ $c \leftarrow a$

- ✓ a et b = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
c	Caractère	Variable auxiliaire.

2) Algorithme :

- 0) Procédure Permut (VAR a, b: Caractère)

1) $c \leftarrow a$

$a \leftarrow b$

$b \leftarrow c$

- 2) Fin Permut

*** Procédure : Affiche ***

1) Spécification du problème :

Résultat = Aff

Traitements =

✓ Aff = []

↵ Afficher (T [i])

↵ Si $i < n$ alors exécuter de nouveau la procédure Affiche en incrémentant la valeur de i par 1.

- ✓ T, n et i = Paramètres formels

2) Algorithme :

- 0) Procédure Affiche (n,i: Entier ; T:TAB)

1) Ecrire (T [i])

2) Si $i < n$

Alors

Affiche (n,i+1 , T)

FinSi

3) Fin Affiche

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

***** Traduction en Turbo Pascal *******Program** Tri;

Uses WinCRT ;

Type

Tab= Array [1..60] of Char;

Var

T: Tab;

n : Integer;

Procedure Saisie (VAR n : Integer);

Begin

Write ('Saisir le nombre de caractères à introduire : ');

Readln (n) ;

If Not (n In [2..60])

Then

Saisie (n);**End;****Procedure** Remp (n : Integer ; VAR T : TAB);

Begin

Write ('Saisir le caractère n °', n, ' : ');

Readln (T[n]);

If Not (ORD (T[n]) In [33..127])

Then

Remp (n, T)

Else

If n>1 Then **Remp (n-1, T);****End;****Procedure** Affiche (n, i: Integer ; T:TAB) ;

Begin

Write (T [i], ' ');

If i<n

Then

Affiche (n, i+1, T);**End;**

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Procedure Permut (VAR a, b: Char);

Var c : Char ;

Begin

 c:=a;

 a:=b;

 b:=c;

End;

Procedure Trier (n,i: Integer; VAR T: TAB; v : boolean) ;

Begin

 If T[i] < T[i+1]

 Then

 begin

 Permut (T[i], T[i+1]) ;

 v := false;

 end;

 If i<n-1

 Then

Trier (n, i+1, T, v)

 Else

 If v = false

 Then

Trier (n, 1, T, True);

End;

{Programme principal}

Begin

 Saisie (n);

 Remp (n, T);

 Trier (n, 1, T, True);

 Clrscr;

 Writeln ("*** Le contenu de T trié dans l'ordre décroissant : ***");

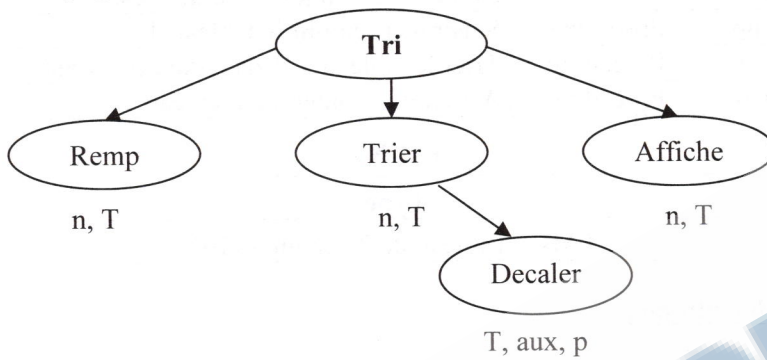
 Affiche (n, 1, T);

End.

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°4 :

I- Découpage



- ✓ **Tri** : programme principal, qui permet d'afficher le contenu de T trié dans l'ordre croissant.
- ✓ **Remp** : procédure, permettant de saisir le nombre de chaînes de caractères n et de remplir le tableau T.
- ✓ **Trier** : procédure, permettant de trier le tableau T dans l'ordre croissant.
- ✓ **Decaler** : procédure, permettant de décaler toutes les chaînes de caractères supérieures à **aux** vers la droite et de déterminer la position d'insertion.
- ✓ **Affiche** : procédure, permettant d'afficher le contenu du tableau T.

II- Analyses & Algorithmes

*** Programme principal : Tri ***

1) Spécification du problème :

Résultat = Affiche (n, T)

Traitements =

- ✓ n, T = Remp (n, T)
- ✓ T = Trier (n, T)
- ✓ Affiche, Remp et Trier = Procédures

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
n	Entier	Contenir le nombre de chaînes de caractères.
T	TAB	Contenir n chaînes de caractères.
Remp	Procédure	Saisir n et remplir le tableau T.
Trier	Procédure	Trier le tableau T dans l'ordre croissant.
Affiche	Procédure	Afficher le contenu du tableau T.

T.D.N.T

Type
TAB = Tableau de 20 chaînes [10]

2) Algorithme :

- 0) Début Tri
- 1) Remp (n, T)
- 2) Trier (n, T)
- 3) Affiche (n, T)
- 4) Fin Tri

*** Procédure : Remp ***

1) Spécification du problème :

Résultat = n, T

Traitements =

- ✓ n = Donnée avec $n \in [4..20]$
- ✓ T = Données
 - ↳ T[i] = Donnée avec chaque T[i] est de taille $\in [2.. 10]$.
- ✓ i = Compteur allant de 1 à n
- ✓ n et T = Paramètres formels

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) Algorithme :

- 0) Procédure Remp (VAR n : Entier ; VAR T : TAB)
- 1) Répéter
 - Ecrire ("Saisir le nombre de chaînes de caractères: ")
 - Lire (n)
 - Jusqu'à n Dans [4..20]

- 2) Pour i de 1 à n Faire
 Répéter
 Ecrire ("Saisir la chaîne n° ", i , " : ")
 Lire ($T[i]$)
 Jusqu'à Long ($T[i]$) Dans [2.. 10]
 FinPour
 3) Fin Remp



*** Procédure : Trier ***

1) Spécification du problème :

Résultat = T

Traitements =

- ✓ $T = [\quad]$
- ↻ Pointer la chaîne de caractères n° i afin de la comparer avec les $i-1$ chaînes précédentes se trouvant dans T.
- ↻ Si la chaîne pointée est plus petite que la précédente il faut :
 - Sauvegarder la chaîne pointée dans une case auxiliaire ($aux \leftarrow T[i]$),
 - Fixer la position de départ de décalage dans la position p ($p \leftarrow i-1$),
 - Appeler la procédure Décaler assurant les décalages possibles à droite,
 - Le dernier décalage se termine à une position (p), la position p peut être = 0 si toutes les $(i-1)$ chaînes sont supérieures à la $i^{\text{ème}}$ chaîne où à une position dans le tableau indiquant que la $p^{\text{ème}}$ chaîne est inférieure à la $i^{\text{ème}}$ chaîne,
 - Pour les deux valeurs possibles de p il faut insérer le contenu de (aux) à la position de la dernière chaîne décalée ($T[p+1] \leftarrow Aux$).
- ✓ i = Compteur allant de 2 à n
- ✓ n et T = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur.
P	Entier	Compteur.
Aux	Chaîne [10]	Variable intermédiaire.
Décaler	Procédure	Décaler à droite des chaînes de caractères.

2) Algorithmme :

0) Procédure Trier (n: Entier ; VAR T: TAB)

1) Pour i de 2 à n Faire

 Si $T[i] < T[i-1]$

Alors

 Aux $\leftarrow T[i]$ $p \leftarrow i - 1$

Decaler (T, aux, p)

 $T[p+1] \leftarrow \text{Aux}$

FinSi

FinPour

2) Fin Trier

***** Procédure : Decaler *******1) Spécification du problème :****Résultat = T****Traitements =**

- ✓ $T = [\quad]$
 - ↳ $T[p+1] \leftarrow T[p]$
 - ↳ $p \leftarrow p - 1$ décrémente p par 1
 - ↳ Répéter les deux tâches précédentes jusqu'à $T[p] < \text{aux}$ ou $p = 0$
- ✓ p, aux et T = Paramètres formels

2) Algorithmme :

0) Procédure Decaler (Var T: TAB; aux : chaîne ; Var p : entier;)

1) Répéter

 $T[p+1] \leftarrow T[p]$ $p \leftarrow p - 1$ Jusqu'à ($T[p] < \text{aux}$) OU ($p = 0$)

2) Fin Decaler

 $T[p+1] \leftarrow T[p]$

Décalage à droite des éléments du tableau T pour permettre l'insertion du nouvel élément.

***** Procédure : Affiche *******1) Spécification du problème :****Résultat = Aff****Traitements =**

- ✓ $\text{Aff} = [\quad]$
 - ↳ Afficher (T [i])
- ✓ i = Compteur allant de 1 à n
- ✓ n et T = Paramètres formels

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur.

2) **Algorithme :**

- 0) Procédure Affiche (n: Entier ; T:TAB)
- 1) Ecrire ("Le contenu de T trié dans l'ordre croissant")
 - Pour i de 1 à n Faire
 - Ecrire (T [i])
 - FinPour
- 2) Fin Affiche

Exercice n°5 :**Program** Tri;

Uses WinCRT;

Type Tab = Array [1..20] of String[10];**Var**

T : Tab;

n : integer;

Procedure Saisie (Var n : integer);

Begin

Writeln ('Saisir la taille du tableau : ');

Readln (n);

If Not (n in [4..20])

Then **Saisie (n);****End;****Procedure** Remp (Var T:Tab ; n : integer);

Begin

Writeln ('Taper la chaîne n° ',n, ' : ');

Readln (T[n]);

If **Not** (Length (T[n]) In [2..10])

Then

Remp (T,n)

Else

If n>1

Then **Remp (T, n-1);****End;**

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

Procedure Decaler (Var T: Tab ; aux: string; Var p : integer);
Begin

T[p+1]:=T[p];

p := p-1;

If (T[p]>Aux) and (P>0)

Then **Decaler (T,aux, p);**

End;

Procedure Trier (Var T:Tab ; n, i : integer);

Var

aux :string;

p: integer;

Begin

If T[i]<T[i-1] then

begin

aux :=T[i];

p:=i-1;

Decaler (T, aux, p);

T [p+1]:=aux;

End;

If (i<n) Then

Trier (T, n, i+1);

End;

Procedure Affiche (T : Tab ; n, i : integer);

Begin

Write (T[i], '');

If (i<n)

Then **Affiche (T, n, i+1);**

End;

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

{Programme principal}

Begin

Saisie (n);

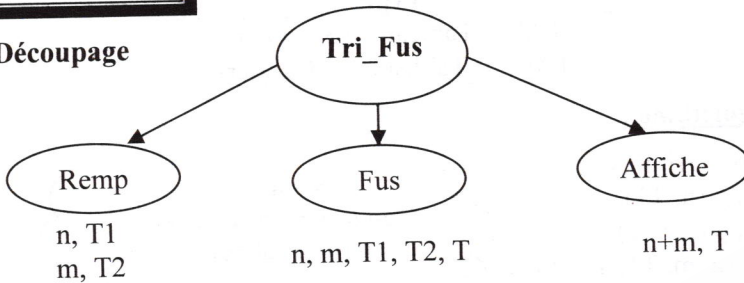
Remp (T, n);

Trier (T, n, 2);

Writeln ('Le contenu de T trié dans l'ordre croissant est : ');

Affiche (T, n, 1);

End.

Exercice n°6 :**I- Découpage**

- ✓ **Tri_Fus** : programme principal, qui permet de fusionner T1 et T2 dans un 3^{ème} tableau T pour obtenir un tableau trié.
- ✓ **Remp**: procédure, permettant de saisir n et remplir un tableau T par n entiers triés dans l'ordre décroissant.
- ✓ **Fus** : procédure, permettant de fusionner T1 et T2 pour obtenir un tableau T trié.
- ✓ **Affiche** : procédure, permettant d'afficher le contenu d'un tableau.

II- Analyses & Algorithmes

*** Programme principal : Tri_Fus ***

1) Spécification du problème :

Résultat = Affiche (n+m, T)

Traitements =

- ✓ T = Fus (n, m, T1, T2, T)
- ✓ n, T1 = Remp (n, T1)
- ✓ m, T2 = Remp (m, T2)
- ✓ Remp, Fus et Affiche = modules

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
n	Entier	Contenir le nombre d'entiers de T1.
T1	TAB	Contenir n entiers.
m	Entier	Contenir le nombre d'entiers de T2.
T2	TAB	Contenir m entiers.
T	TABF	Contenir n+m entiers.
Affiche	Procédure	Afficher les n+ m entiers de T.
Fus	Procédure	Fusionner les deux tableaux T1 et T2.
Remp	Procédure	Saisir le nombre d'entiers et remplir un tableau.

T.D.N.T

Type
TAB = Tableau de 30 entiers
TABF = Tableau de 60 entiers

2) Algorithme :

- 0) Début Tri_Fus
- 1) Remp (n, T1)
- 2) Remp (m, T2)
- 3) Fus (n, m, T1, T2, T)
- 4) Affiche (n + m, T)
- 5) Fin Tri_Fus

n + m !?
C'est le nombre d'entiers
du nouveau tableau T.

*** Procédure : Remp ***

1) Spécification du problème :

Résultat = x, V

Traitements =

- ✓ x = Donnée avec $x \in [3..30]$
- ✓ V = Données
 - ↳ V[1] = Donnée
 - ↳ V[i] = Donnée avec $V[i] \leq V[i-1]$
- ✓ i = Compteur allant de 2 à x
- ✓ x et V = Paramètres formels

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) Algorithme :

- 0) Procédure Remp (**Var** x : Entier ; **Var** V : TAB)
- 1) Répéter
 - Ecrire ("Saisir le nombre d'entiers à entrer : ") ; Lire (x)
 - Jusqu'à x Dans [3..30]
- 2) Ecrire ("Saisir l'entier n ° 1 : ") ; **Lire** (V[1])
 - Pour i de 2 à x Faire
 - Répéter
 - Ecrire ("Saisir l'entier n °", i, " : ")
 - Lire (V[i])
 - Jusqu'à $V[i] \leq V[i-1]$
 - FinPour
- 3) Fin Remp

*** Procédure : Affiche***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
- ↳ Afficher (V [i])
- ✓ i = Compteur allant de 1 à x
- ✓ x et V = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) Algorithme :

- 0) Procédure Affiche (x: Entier; V:TABF)
- 1) Pour i de 1 à x Faire
 - Ecrire (V [i])
 - FinPour
- 2) Fin Affiche

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Fus ***

1) Spécification du problème :

Résultat = T

Traitements =

- ✓ T = [i ← 0 ; j ← 1 ; k ← 1]
- a) i ↗ +1
- b) T[i] ← T1[j] et j ↗ +1 si T1[j] > T2[k]
- c) Sinon T[i] ← T2[k] et k ↗ +1
- ↳ Répéter les traitements a, b et c jusqu'à (j > n) ou (k > m)
- ↳ Affecter le reste des entiers de T1 dans T si k > m :
 - T[i] ← T1[r] et i ↗ +1
 - r = Compteur allant de j à n
- ↳ Sinon Affecter le reste des entiers de T2 dans T :
 - T[i] ← T2[r] et i ↗ +1
 - r = Compteur allant de k à m
- ✓ n, m, T1, T2 et T = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur
j	Entier	Compteur
k	Entier	Compteur
r	Entier	Compteur

2) **Algorithme :**

0) Procédure Fus (n, m: Entier ; T1, T2: TAB ; VAR T: TABF)

1) $i \leftarrow 0$; $j \leftarrow 1$; $k \leftarrow 1$

Répéter

 $i \leftarrow i + 1$ Si $T1[j] > T2[k]$

Alors

 $T[i] \leftarrow T1[j]$ $j \leftarrow j + 1$

Sinon

 $T[i] \leftarrow T2[k]$ $k \leftarrow k + 1$

FinSi

Jusqu'à ($j > n$) ou ($k > m$)Si $k > m$ Alors

Pour r de j à n Faire

 $i \leftarrow i + 1$ $T[i] \leftarrow T1[r]$

FinPour

FinSi

Si $j > n$ Alors

Pour r de k à m Faire

 $i \leftarrow i + 1$ $T[i] \leftarrow T2[r]$

FinPour

FinSi

2) Fin Fus

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Exercice n°7:

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Program Tri_shell;

Uses WinCRT;

Type

Tab = Array [1..100] of Real;

Var

T : Tab;

n : Integer;

Procedure Remp (VAR n : Integer; VAR T : TAB);

Var

i : Integer ;

Begin

Repeat

Write ('Saisir le nombre de valeurs à traiter : ');

Readln (n);

Until n In [10..100];

Randomize;

For i := 1 To n Do

T[i]:= **Random;**

Remplir T par n réels aléatoires.

End;**Procedure** Affiche (n: Integer; T:TAB) ;

Var

i : Integer ;

Begin

Writeln ('Le nouveau contenu du tableau T');

For i := 1 To n Do

Write (T [i]:2:3, ' ');

End;**Procedure** Tri (n: Integer; Var T:TAB);

Var

i, p, j : Integer ;

aux : Real;

Begin

P := 0;

While p<n Do

p:=(3*p+1);

Calcul du plus grand pas de décalage possible (p).
Ce n'est qu'une initialisation du pas (p).

While $p < 0$ Do

Begin

$p := (p \text{ div } 3);$

For $i := p$ To n Do

Begin

$aux := T[i];$

$j := i;$

While $(j > p - 1)$ And $(T[j - p] > aux)$ Do

Begin

$T[j] := T[j - p];$

$j := j - p;$

End;

$T[j] := aux;$

End;

End;

End;

{Programme principal}

Begin

Remp (n, T);

Tri (n, T);

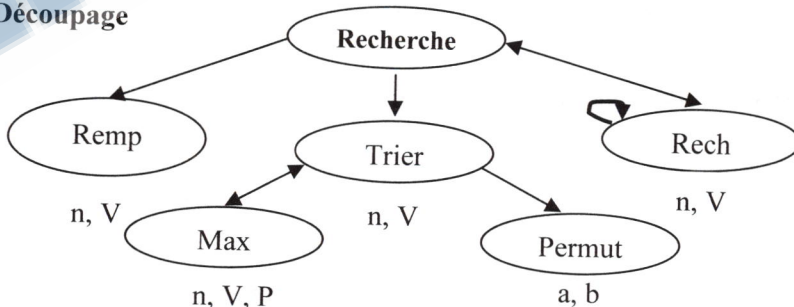
Affiche (n, T);

End.

Exercice n°8 :

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

I- Découpage



Recherche : programme principal, qui permet de rechercher d'une façon rapide l'existence d'un entier x dans un vecteur V .

- ✓ **Remp** : procédure, permettant de saisir n et de remplir le vecteur V .
- ✓ **Trier** : procédure, permettant de trier le vecteur v dans l'ordre croissant, pour permettre la recherche dichotomique de l'entier x .
- ✓ **Rech** : fonction **réursive**, retournant la vérité de l'existence de x dans V .
- ✓ **max** : fonction, retournant la position du plus grand entier se trouvant dans le vecteur V à partir d'une position P .
- ✓ **Permut** : procédure, permettant de permuter deux entiers a et b .

II- Analyses & Algorithmes

*** Programme principal : Recherche ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
- ↻ Afficher ("La valeur recherchée existe.") Si Rech ($V, 1, n, x$) = Vrai
- ↻ **Sinon** Afficher ("La valeur recherchée n'existe pas")
- ✓ x = donnée avec x est un entier pair
- ✓ n, V = Remp (n, V)
- ✓ V = Trier (n, V)
- ✓ Remp, Trier et Rech = Modules

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U.G

Objet	Nature/Type	Rôle
x	Entier	Contenir la valeur à rechercher dans V .
n	Entier	Contenir le nombre d'entiers à saisir dans V .
V	VECT	Contenir n entiers pairs.
Remp	Procédure	Saisir la valeur de n et remplir le vecteur V .
Rech	Fonction/ Booléen	Verifier l'existence d'un entier x dans V .
Trier	Procédure	Trier le vecteur V dans l'ordre croissant.

T.D.N.T

Type
VECT = Tableau de 2000 entiers

2) Algorithme :

0) Début Recherche

1) Remp (n, V)

2) Trier (n, V)

3) Répéter

Ecrire ("Saisir la valeur à rechercher : ")

Lire (x)

 Jusqu'à $x \text{ MOD } 2 = 0$

4) Si Rech (V, 1, n, x)

Alors Ecrire ("La valeur recherchée existe.")

Sinon Ecrire ("La valeur recherchée n'existe pas!")

FinSi

5) Fin Recherche

***** Procédure : Remp *******1) Spécification du problème :****Résultat** = n, V**Traitements** =✓ n = Donnée avec $n \in [10..2000]$

✓ V = Données

 ↳ $V[i] = \text{Donnée avec } V[i] \text{ Mod } 2 = 0$

✓ i = Compteur allant de 1 à n

✓ n et V = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) Algorithme :

0) Procédure Remp (VAR n : Entier ; VAR V : VECT)

1) Répéter

Ecrire ("Saisir le nombre d'entiers à introduire : ")

Lire (n)

 Jusqu'à $(n \geq 10)$ ET $(n \leq 2000)$

2) Pour i de 1 à n Faire

Répéter

Ecrire ("Saisir l'entier n °", i, " : ")

Lire (V[i])

 Jusqu'à $V[i] \text{ Mod } 2 = 0$

FinPour

3) Fin Remp

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

***** Procédure : Trier *******1) Spécification du problème :****Résultat** = V**Traitements** =

- ✓ $V = [\quad]$
- ↪ Rechercher la position (P) du plus grand entier en utilisant Max (n,i, V)
- ↪ Si (P) est différente du numéro de la case pointée (i) alors on permutera les valeurs de V[P] et de V[i] en utilisant Permut (V[i], V[P])
- ✓ i = Compteur allant de 1 à n-1
- ✓ Max et Permut = Modules
- ✓ n et V = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur.
P	Entier	Contenir la position du plus grand entier.
Max	Fonction / Entier	Retourner la position du plus grand entier se trouvant dans V à partir de la position P.
Permut	Procédure	Permuter deux entiers a et b.

2) Algorithme :

0) Procédure Trier (n : Entier; VAR V: VECT)

1) Pour i de 1 à n-1 Faire

P ← Max (n, P, V)

Si P > i Alors Permut (V[i], V[P]) FinSi

FinPour

2) Fin Trier

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

***** Procédure : Permut *******1) Spécification du problème :****Résultat** = a, b**Traitements** =

- ✓ a, b = [\quad]
- ↪ c ← a
- ↪ a ← b
- ↪ b ← c
- ✓ a et b = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
c	Entier	Variable intermédiaire.

2) Algorithme :

0) Procédure Permut (VAR a, b: Entier)

1) $c \leftarrow a$ $a \leftarrow b$ $b \leftarrow c$

2) Fin Permut

***** Fonction : Max *******1) Spécification du problème :****Résultat** = Max \leftarrow m**Traitements** =✓ $m = [m \leftarrow \text{Debut} \quad]$
 \Downarrow $m \leftarrow i$ Si $V[i] > V[m]$ ✓ $i =$ Compteur allant de (Debut +1) à n

✓ Debut, n et V = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
m	Entier	Variable intermédiaire.
i	Entier	Compteur.

2) Algorithme :

0) Fonction Max (n, Debut: Entier ; V: VECT): Entier

1) $m \leftarrow \text{Debut}$

Pour i de (Debut+1) à n Faire

 Si $V[i] > V[m]$

Alors

 $m \leftarrow i$

FinSi

FinPour

2) Max \leftarrow m

3) Fin Max

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Fonction : Rech ***

1) Spécification du problème :

Résultat = Rech

Traitements =

✓ R =

↪ $R \leftarrow \text{Vrai}$ Si $V[m] = x$ ↪ $R \leftarrow \text{Faux}$ Si $d > f$ ↪ $R \leftarrow \text{Rech}(V, d, m-1, x)$ Si $V[m] > x$ (V est trié dans l'ordre croissant)↪ $R \leftarrow \text{Rech}(V, m+1, f, x)$ Si $V[m] < x$ ✓ $m \leftarrow (d + f) \text{ DIV } 2$

✓ d, f, x et V = Paramètres formels

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

2) Algorithme :

0) Fonction Rech (V: VECT; d, f, x : Entier): Booléen

1) $m \leftarrow (d + f) \text{ DIV } 2$ 2) Si $V[m] = x$

Alors

Rech \leftarrow Vrai

Sinon

Si $d > f$

Alors

Rech \leftarrow Faux

Sinon

Si $V[m] > x$

Alors

Rech \leftarrow Rech (V, d, m-1, x)

Sinon

Rech \leftarrow Rech (V, m+1, f, x)

FinSi

FinSi

FinSi

3) Fin Rech

L'arrêt de la fonction Rech est défini suite à l'un des deux cas suivants:
1- La valeur recherchée est trouvée,
2- Tous les entiers de V sont parcourus.

Continuer la recherche mais dans la 1^{ère} moitié de V en gardant l'indice minimal de départ de la recherche (d) et en modifiant l'indice maximal de la fin de la recherche (f) par (m-1).

Continuer la recherche mais dans la 2^{ème} moitié de V en gardant l'indice maximal de la fin de la recherche (f) et en modifiant l'indice minimal de départ de la recherche (d) par (m+1).

Exercice n°9:**Program** Tri_Fichier;

Uses WinCRT;

TypeElevé = **Record**

Nom, Pren : String[20];

Moy : Real;

End;

Fich = **File** of Elevé;**Var**

f : Fich;

Procedure Saisie (VAR x : Elevé);

Begin

With x Do

Begin

Repeat

Write ('Nom: ');

Readln (Nom);

Until Nom [1] In ['A'..'Z'];

Repeat

Write ('Prénom: ');

Readln (Pren);

Until Pren [1] In ['A'..'Z'];

Repeat

Write ('Moyenne: ');

Readln (Moy);

Until (Moy >= 0) And (Moy <= 20);

End;**End;****Procedure** Creation (Var f: Fich);

Var

x : Elevé;

n, i: integer;

Begin

Rewrite (f);

Repeat

Write ('Nombre d"élèves : ');

Readln (n);

Until n In [1..40];

Saisie des données d'un élève.

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

```
For i:= 1 to n do
  Begin
```

```
    Saisie (x);
    Write (f, x);
```

```
  End;
```

```
End;
```

Remplissage du fichier par les données d'une classe de n élèves.

```
Procedure Permut (Var f : Fich; x,y : Eleve; p : Word);
```

```
Begin
```

```
  Reset (f);
```

```
  Seek (f, p-2);
```

```
  Write (f, y);
```

```
  Write (f, x);
```

```
End;
```

Ce n'est pas le même principe de la permutation de deux entiers ou de deux cases d'un tableau ! C'est un jeu d'échange de places : y est écrit sur l'élément source de x et x est écrit sur l'élément source de y.

```
Procedure Tri (Var f : Fich);
```

```
Var
```

```
  V : Boolean;
```

```
  x, y : Eleve;
```

```
Begin
```

```
  Repeat
```

```
    V := True;
```

```
    Reset (f);
```

```
    While Not (Filepos (f) = Filesize (f)-1) Do
```

```
      Begin
```

```
        Read (f,x);
```

```
        Read (f,y);
```

```
        If (y.Nom <x.Nom) Or ((y.Nom = x.Nom) And (y.Pren <x.Pren))
```

```
          Then
```

```
            Begin
```

```
              Permut (f, x, y, Filepos (f));
```

```
              v:=False;
```

```
            end;
```

```
            Seek (f, Filepos (f)-1);
```

```
          end;
```

```
        Until V;
```

```
End;
```

Parcourir (f) du 1^{er} élément jusqu'à l'avant dernier élément.

Réajuster la position du pointeur, pour ne pas sauter quelques éléments.

```

Procedure Affiche (Var f : Fich);
Var x : Eleve;
Begin
  Reset (f);
  Writeln ('N°   Nom           Prénom           Moyenne');
  While Not (EOF(f)) Do
  Begin
    Read (f, x);
    With x Do
      Write (Filepos (f), '   ', Nom, ' ', Pren, ' ', Moy:2:2);
    Writeln;
  end;
End;

```

Le rang de l'élève.
 {Programme principal}

Begin

```

Assign (f, 'C:\Eleves.FCH');
Writeln ('*** Création du fichier des élèves *** ');
Creation (f);
Tri (f);
Writeln ('*** Tous les élèves triés dans l'ordre croissant *** ');
Affiche (f);

```

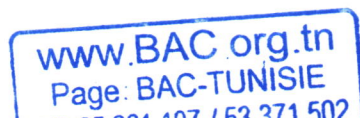
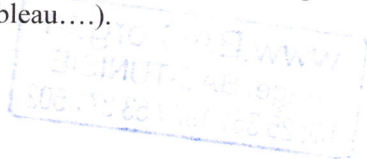
End.

Remarques :

La méthode de tri utilisée pour la résolution de cet exercice est celle du tri à bulles. Comme vous le remarquez, le principe est le même que celui du tri d'une liste de valeurs se trouvant dans un tableau ou d'une liste de caractères d'une chaîne de caractères au niveau:

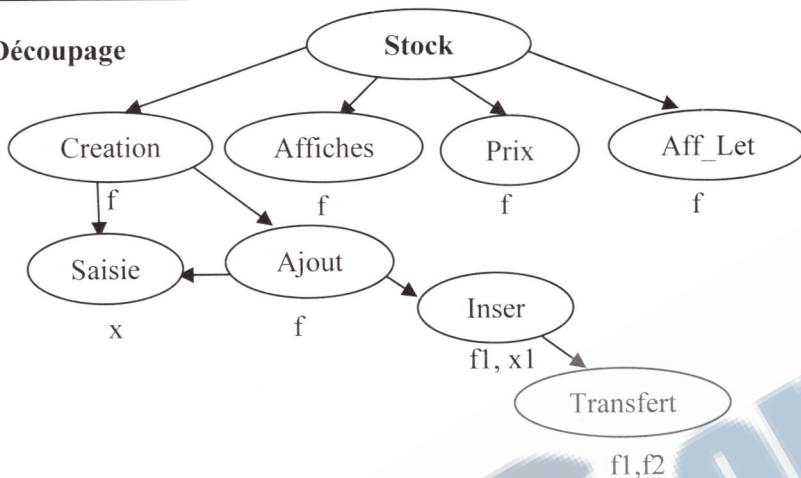
- ✓ De l'accès et du parcours des éléments,
- ✓ De la gestion des compteurs,
- ✓ Des comparaisons entre les éléments,
- ✓ De la permutation ou l'échange de place,

Avec le même raisonnement et une simple conversion de commandes et d'instructions, vous pouvez réussir n'importe quelle méthode de tri sur n'importe quelle structure de donnée (fichier, tableau....).



Exercice n°10:

I- Découpage



- ✓ **Stock** : programme principal, qui permet de gérer le stock d'un revendeur.
- ✓ **Creation** : procédure, permettant de créer un fichier contenant le stock des articles triés dès la saisie.
- ✓ **Saisie** : procédure, permettant de saisir les données d'un article.
- ✓ **Ajout** : procédure, permettant d'ajouter un article dans le fichier en gardant le tri du fichier.
- ✓ **Inser** : procédure, permettant d'ajouter des articles dans un fichier en le gardant trié.
- ✓ **Transfert** : procédure, permettant de transférer les éléments du fichier intermédiaire f2 au fichier principal f1.
- ✓ **Aff_Let** : procédure, permettant d'afficher les désignations des articles commençant par une lettre donnée.
- ✓ **Prix** : procédure, permettant d'afficher les articles dont le prix unitaire se trouve dans une marge de prix donnée.
- ✓ **Affiches** : procédure, permettant d'afficher tous les articles du fichier.

II- Analyses & Algorithmes

*** Programme principal : Stock ***

1) Spécification du problème :

Résultat = Aff

Traitements =

✓ Aff = []

↳ Affiches (f)

↳ Aff_Let(f)

↳ Prix (f)

✓ f = Creation (f)

✓ f = Assigner (f, "C:\Stock.FCH")

✓ Creation, Prix, Aff_Let et Affiches = Modules

T.D.O.U.G

Objet	Nature/Type	Rôle
f	Fich	Contenir des données d'un Stock d'accessoires.
Affiches	Procédure	Afficher tous les articles du fichier.
Aff_Let	Procédure	Afficher les articles dont la désignation commence par une lettre donnée.
Prix	Procédure	Afficher les articles dont le prix unitaire se trouve dans une marge de prix donnée.
Creation	Procédure	Remplir un fichier f par les données d'un stock d'accessoires d'automobiles.

T.D.N.T

Type
Fich = Fichier de Art
Art = Enregistrement
NUM : Mot
DES : Chaîne [25]
PU : Réel
Fin Art

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

2) Algorithme :

- 0) Début Stock
- 1) Assigner (f, "C:\Stock.FCH")
- 2) Creation (f)
- 3) Affiches (f)
- 4) Aff_Let (f)
- 5) Prix (f)
- 6) Fin Stock

***** Procédure : Creation *******1) Spécification du problème :****Résultat = f****Traitements =**

- ✓ f = []
 - ↳ Ecrire (f, x)
 - ↳ x = Saisie (x)
 - ↳ Ajout (f)
- ✓ f = Recréer (f)
- ✓ Ajout et Saisie = Modules
- ✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Art	Variable intermédiaire
Ajout	Procédure	Insérer x dans une position bien définie de f
Saisie	Procédure	Saisir un article

2) Algorithme :

- 0) Procédure Creation (VAR f: Fich)
 - 1) Recréer (f)
 - 2) Saisie (x)
Ecrire (f, x)
 - 3) Ajout (f)
 - 4) Fin Creation

*** Procédure : Saisie ***

1) Spécification du problème :

Résultat = x

Traitements =

- ✓ x = Données
- ↪ x.Num = Sera défini dans la procédure **Inser** (Voir la page suivante),
- ↪ x.Des = Donnée dont le 1^{er} caractère est une lettre en majuscule,
- ↪ x.PU = Donnée avec $X.PU > 0$,
- ✓ x = Paramètre formel.

2) Algorithme :

0) Procédure Saisie (VAR x : Art)

1) Avec x Faire

Répéter

Ecrire ("Désignation : ")

Lire (Des)

Jusqu'à Des[1] Dans ["A".."Z"]

Répéter

Ecrire ("Prix unitaire : ")

Lire (PU)

Jusqu'à PU > 0

FinAvec

2) Fin Saisie

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Ajout ***

1) Spécification du problème :

Résultat = f

Traitements =

- ✓ f = []
- ↪ Inser (f, x)
- ↪ x = Saisie (x)
- ↪ Répéter ces deux dernières tâches jusqu'à Majus (Rep) = "N"
- ↪ Rep = Donnée avec Majus (Rep) ∈ ["O", "N"]
- ✓ f = Ouvrir (f)
- ✓ Inser et Saisie = Procédures
- ✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Art	Variable intermédiaire.
Rep	Caractère	Contenir une confirmation de l'utilisateur.
Saisie	Procédure	Saisir un article.
Inser	Procédure	Insérer x à une position précise dans f.

2) Algorithme :

0) Procédure Ajout (VAR f : Fich)

1) Ouvrir (f)

2) Répéter

Saisie (x)

Inser (f, x)

Répéter

Ecrire ("Autre article (O/N) ? : ")

Lire (Rep)

Jusqu'à Majus (Rep) Dans ["O", "N"]

Jusqu'à Majus (Rep) = "N"

3) Fin Ajout

*** Procédure : Inser ***

1) Spécification du problème :

Résultat = f1

Traitements =

✓ f1 = Transfert (f1, f2)

✓ f1 = Ouvrir (f1)

✓ f1 = Paramètre formel

✓ f2 = []

↷ Ecrire (f2, x2)

↷ Lire (f1, x2)

} Copier tous les x2 de f1, dont le champ désignation (x2.DES) est inférieur à celui de x1, dans un 2^{ème} fichier intermédiaire f2.

↷ Reculer le pointeur de f1, d'un article s'il n'est pas copié dans f2

↷ Ecrire (f2, x1) insérer le nouvel article

↷ Compléter le reste des éléments de f1 dans f2

↷ Chaque écriture dans f2 doit être précédée par une instruction de mise à jour du champ num $x2.num \leftarrow Position_fichier(f2)+1$ et ce sera la même chose avec l'élément à insérer : $x1.num \leftarrow Position_fichier(f2) + 1$

- ✓ $x2 = \text{Lire}(f1, x2)$
- ✓ $f2 = \text{Recréer}(f2)$
- ✓ $f2 = \text{Assigner}(f2, "C:\text{Stock2.FCH}")$
- ✓ $\text{Transfert} = \text{Procédure}$
- ✓ $x1 = \text{Paramètre formel}$

T.D.O.U.L

Objet	Nature / Type	Rôle
f2	Fich	Fichier intermédiaire.
x2	Art	Variable intermédiaire.

2) **Algorithme :**

0) Procédure Inser (Var f1 : Fich ; x1 : Art)

1) Assigner (f2, "C:\Stock2.FCH")

Recréer (f2)

Ouvrir (f1)

2) Répéter

Lire (f1, x2)

Si $x2.Des < x1.Des$

Alors

 $x2.Num \leftarrow \text{Position_fichier}(f1)$

Ecrire (f2, x2)

Sinon

Rechercher_position (f1, Position_fichier (f1)-1)

FinSi

Jusqu'à (FF(f1)) OU $(x2.Des > x1.Des)$ $x1.Num \leftarrow \text{Position_fichier}(f2)+1$

Ecrire (f2, x1)

Tantque (Non (FF(f1))) Faire

Lire (f1, x2)

 $x2.Num \leftarrow \text{Position_fichier}(f2) + 1$

Ecrire (f2, x2)

FinTantque

3) Transfert (f1, f2)

4) Fin Inser

www.BAC.org.tn
 Page BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

*** Procédure : Transfert ***

1) Spécification du problème :

Résultat = f1

Traitements =

- ✓ f1 = []
- ↳ Copier tous les x du fichier intermédiaire f2 dans le fichier principal f1,
- ↳ Ecrire (f1, x)
- ✓ x = Lire (f2, x) tant qu'il y a des articles dans le fichier f2.
- ✓ f1 = Recréer (f1)
- ✓ f2 = Ouvrir (f2)
- ✓ f1 et f2 = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Art	Variable intermédiaire.

2) Algorithme :

- 0) Procédure Transfert (Var f1, f2 : Fich)
- 1) Recréer (f1)
- 2) Ouvrir (f2)
- 3) Tantque Non (FF(f2)) Faire
 - Lire (f2, x)
 - Ecrire (f1, x)
 FinTantque
- 4) Fin Transfert

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

*** Procédure : Prix ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
- a) Ecrire (x.Des) si (x.PU >= pu1) ET (x.PU <= pu2)
- b) x = Lire (f, x)
- ↳ Répéter les tâches a et b tant qu'il y a des articles dans le fichier f.

- ✓ $pu1$ = Donnée avec $pu1 \geq 0$,
- ✓ $pu2$ = Donnée avec $pu2 > pu1$,
- ✓ f = Ouvrir (f)
- ✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
$pu1$	Réel	Contenir la borne inférieure des prix unitaires à rechercher
$pu2$	Réel	Contenir la borne supérieure des prix unitaires à rechercher
x	Art	Variable intermédiaire

2) **Algorithme :**

- 0) Procédure Prix (Var f : Fich)
- 1) Ouvrir (f)
- 2) Répéter
 - Ecrire ("Saisir la borne inférieure des prix unitaires : ")
 - Lire ($pu1$)
 - Jusqu'à $pu1 \geq 0$
- 3) Répéter
 - Ecrire ("Saisir la borne supérieure des prix unitaires : ")
 - Lire ($pu2$)
 - Jusqu'à $pu2 > pu1$
- 4) Tantque Non (FF(f)) Faire
 - Lire (f, x)
 - Si ($x.PU \geq pu1$) ET ($x.PU \leq pu2$)
 - Alors
 - Ecrire ($x.Des$)
 - FinSi
 - FinTantque
- 5) Fin Prix

www.BAC.org.tn
 Page: BAC-TUNISIE
 Tél: 25 361 197 / 53 371 502

*** Procédure : Aff_Let ***

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = []
 - a) Ecrire (x.Des) si $x.Des[1] = c$
 - b) $x = Lire(f, x)$
- ↳ Répéter les tâches a et b tant qu'il y a des articles dans le fichier f
- ✓ c = Donnée avec c est une lettre en majuscule,
- ✓ f = Ouvrir (f)
- ✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
c	Caractère	Contenant une lettre, critère de recherche des articles
x	Art	Variable intermédiaire

2) Algorithme :

- 0) Procédure Aff_Let (Var f : Fich)
- 1) Ouvrir (f)
- 2) Répéter
 - Ecrire ("Tapez une lettre en majuscule : ")
 - Lire (c)
 - Jusqu'à c Dans ["A".."Z"]
- 3) Tantque Non(FF(f)) Faire
 - Lire (f, x)
 - Si $x.Des[1] = c$
 - Alors
 - Ecrire (x.Des)
 - FinSi
- FinTantque
- 4) Fin Aff_Let



*** Procédure : Affiches ***

1) Spécification du problème :

Résultat = Aff

Traitements =

✓ Aff = []

a) Ecrire (x.Num, x.Des, x.Pu)

b) x = Lire (f, x)

↳ Répéter les tâches a et b tant qu'il y a des articles dans le fichier f

✓ f = Ouvrir (f)

✓ f = Paramètre formel

T.D.O.U.L

Objet	Nature / Type	Rôle
x	Art	Variable intermédiaire

2) Algorithme :

0) Procédure Affiche (Var f : Fich)

1) Ouvrir (f)

2) Tantque Non (FF(f)) Faire

Lire (f, x)

Avec x Faire

Ecrire (Num, Des, Pu)

FinAvec

FinTantque

3) Fin Affiche



"La victoire sur soi est la plus grande des victoires."

Platon