

Exercice 1

Proposition	Correction
La recherche dichotomique est une méthode de recherche appliqué sur un tableau quelconque .	La recherche dichotomique est une méthode de recherche appliqué sur un tableau trié . <i>ou encore:</i> La recherche séquentielle est une méthode de recherche appliqué sur un tableau quelconque .
Un tableau est une structure de données permettant de regrouper plusieurs éléments de types différents .	Un tableau est une structure de données permettant de regrouper plusieurs éléments de même type .
Les paramètres formels sont utilisés lors de l' appel d'un sous-programme.	Les paramètres formels sont utilisés lors de la déclaration d'un sous-programme. <i>ou encore:</i> Les paramètres effectifs sont utilisés lors de l' appel d'un sous-programme.
L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de gauche à droite ou de droite à gauche .	L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de gauche à droite .
Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini .	Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs désignées par des identificateurs .

Exercice 2

a.

TDO	
ch, ch1	Chaîne
ch2	Chaîne <i>ou Caractère</i> (avant de voir la question c.)
i, p	Entier

b. .Instruction 2) `ch ← sous_chaine(ch1, 1, p-1)`

.Instruction 4) `ch ← concat(ch, sous_chaine(ch1, p, long(ch1)-p+1))`

ou encore:

.Instruction 4) `insère(ch, long(ch)+1, sous_chaine(ch1, p, long(ch1)-p+1))`

ou encore:

.Instruction 4) `ch ← ch + sous_chaine(ch1, p, long(ch1)-p+1)`

c. `"BAC 2016/2017"`

d. L'algorithme insère **ch2** dans **ch1** à la position **p**

e. `insère(ch2, ch1, p)`

Problème

Analyse Du Programme Principal

Nom: HitParade

Résultat= [Remplir(T1)
 Generer(T2)
 Classer(T1, T2)] Afficher(T1, T2)

Fin HitParade

TDNT	
Tab1	Tableau[1..5] de Chaîne
Tab2	Tableau[1..5] de Entier

TDO	
T1	Tab1
T2	Tab2
Remplir, Generer, Classer, Afficher	Procédure

Procédure Remplir

- 1) Procédure Remplir(Var T1:Tab1)
- 2) Pour i De 1 À 5 Faire
 - Répéter
 - Lire(T1[i])
 - Jusqu'à T1[i] ≠ "" Et Alphanetique(T1[i])
 - Fin Pour
- 3) Fin Remplir

TDO	
Alphanetique	Fonction
i	Octet

Fonction Alphanetique

- 1) Fonction Alphanetique(Ch:Chaîne):Booléen
- 2) i ← 0
- 3) Répéter
 - i ← i+1
 - Jusqu'à Non Majus(Ch[i]) Dans ["A".."Z", " "] Ou i=Long(Ch)
- 4) Alphanetique ← Majus(Ch[i]) Dans ["A".."Z", " "]
- 5) Fin Alphanetique
- 6)

TDO	
i	Octet

Procédure Generer

1) **Procédure** Generer(Var T2:Tab2)

{Lecture du nombre de participants "N"}

2) **Répéter**

 Lire(N)

 Jusqu'à N Dans [5..100]

 {Initialisation de toutes Les cases du tableau T2 à zéro (Les Scores)}

3) **Pour** i De 1 À 5 **Faire**

 T2[i] ← 0

Fin Pour

{Saisie des propositions des participants et Mise à jour des scores des chansons dans T2}

4) **Pour** i De 1 À N **Faire**

Répéter

 Ecrire("Choix du participant N°",i," : ")

 Lire(NumChanson)

 Jusqu'à NumChanson Dans [1..5]

 T2[NumChanson] ← T2[NumChanson] + 3

Fin Pour

5) **Fin** Generer

TDO	
i, N, NumChanson	Octet

Procédure Classer

1) **Procédure** Classer(Var T1:Tab1, Var T2:Tab2)

2) **Répéter**

 NombreDePermutations ← 0

Pour i De 1 À 4 **Faire**

Si T2[i]<T2[i+1] **Alors**

 Kes1 ← T1[i]

 T1[i] ← T1[i+1]

 T1[i+1] ← Kes1

 Kes2 ← T2[i]

 T2[i] ← T2[i+1]

 T2[i+1] ← Kes2

 NombreDePermutations ← NombreDePermutations + 1

Fin Si

Fin Pour

 Jusqu'à NombreDePermutations = 0

3) **Fin** Classer

TDO	
Kes1	Chaîne
NombreDePermutations, Kes2, i	Entier

Procédure Afficher

- 1) Procédure Afficher(T1:Tab1, T2:Tab2)
- 2) Écrire_nl("Le classement est:")
- 3) R ← 1
- 4) Écrire("Rang 1:", T1[1])
- 5) Pour i De 2 À 5 Faire
 - Si T2[i]=T2[i-1] Alors
 - Écrire(", ", T1[i])
 - Sinon
 - R ← R+1
 - Écrire_nl {Retour à La Ligne}
 - Écrire("Rang ",R,":", T1[i])
- Fin Si
- Fin Pour
- 6) Fin Afficher

TDO	
R, i	Octet

