Section : ..... N° d'inscription : ..... Série : .....

Nom et prénom : .....

Date et lieu de naissance : .....

Signatures des surveillants

.....

.....

Épreuve : ***Informatique*** *-* *Sections* ***: Mathématiques, Sciences expérimentales et Sciences techniques*** *-* *Session* ***2024***

20

Le sujet comporte 6 pages numérotées de 1 sur 6 à 6 sur 6.
Les pages de 1 à 4 sont à remettre avec la copie de l'examen.

Exercice N°1 (3 points)

Soit l'algorithme suivant de la fonction **Inverse** permettant de retourner l'entier obtenu en inversant l'ordre des chiffres d'un entier strictement positif ***x*** donné.

Fonction Inverse (x : Entier) : Entier

DEBUT

s ← 0

Tant que (x ≠ 0) Faire

r ← x mod 10

*s ← s * 10 + r*

x ← x div 10

Fin Tant que

Retourner s

FIN

Exemple : Inverse (38921) = 12983

T. D. O. L.

Objet	***Type/Nature***
r, s	*Entier*

1. En exploitant la fonction **Inverse**, compléter la condition de la structure conditionnelle dans l'algorithme de la procédure **Affiche** ci-dessous afin d'afficher si un entier strictement positif ***x*** donné est symétrique ou non.

Un nombre est dit symétrique s'il reste le même lorsqu'il est lu de gauche à droite comme de droite à gauche.

Exemples : **33**, **121** et **4664** sont des entiers symétriques.

Procédure Affiche (x : Entier)

DEBUT

Si …… Alors

Ecrire(x, "est symétrique")

Sinon

Ecrire(x, "n'est pas symétrique")

FinSi

FIN

Ne rien écrire ici

2. Ci-dessous, on donne deux propositions d'algorithme d'une fonction **Inconnue**. Parmi ces propositions, une seule est équivalente à la fonction **Inverse**.

Proposition A

```
Fonction Inconnue (x : Entier) : Entier
DEBUT
    ch ← Convch (x)
    chx ← ""
    Tant que (ch ≠ "") Faire
        chx ← ch[0] + chx
        ch ← Effacer (ch, 0, 1)
    Fin Tant que
    Retourner Valeur (chx)
FIN
```

Proposition B

```
Fonction Inconnue (x : Entier) : Entier
DEBUT
    ch ← Convch (x)
    chx ← ""
    Tant que (ch ≠ "") Faire
        chx ← chx + ch[0]
        ch ← Effacer (ch, 0, 1)
    Fin Tant que
    Retourner Valeur (chx)
FIN
```

T.D.O.L.

Objet	*Type/Nature*
ch, chx	*Chaîne de caractères*

a. Pour chaque proposition d'algorithme, compléter le tableau ci-dessous par le résultat de l'instruction suivante : **Ecrire (Inconnue (6371))**

	Proposition A	**Proposition B**
Résultat	…….	…….

b. En déduire la proposition de l'algorithme équivalente à la fonction **Inverse**.

…….

Voir suite au verso ☞

Ne rien écrire ici

Exercice N°2 (5 points)

Ci-dessous, un algorithme d'une fonction **Existe** permettant de vérifier, à travers une recherche séquentielle, l'existence d'un réel ***a*** dans un tableau ***T*** de ***n*** réels distincts (avec $5 \leq n \leq 100$).

Cette fonction retourne **l'indice** de la case **contenant** le réel ***a*** s'il existe dans ***T*** ou la valeur ***-1*** dans le cas contraire.

N.B. L'indice du premier élément du tableau ***T*** est ***zéro*.**

```
Fonction Existe (a : Réel, T : Tab, n : Entier) : Entier
DEBUT
  i ← -1
  Répéter
     i ← i+1
  Jusqu'à (T[i] = a) Ou (i = n-1)
  Si (T[i] = a) Alors Retourner i
  Sinon Retourner -1
  FinSi
FIN
```

T.D.O.L.

Objet	*Type/Nature*
i	*Entier*

1. Compléter le tableau ci-dessous par la déclaration du nouveau type correspondant au tableau **T**.

T.D.N.T.

Nouveau type
……

2. Ecrire, en utilisant la boucle **Tant que**, l'équivalent de la séquence d'instructions délimitée par des tirets dans l'algorithme de la fonction **Existe**.

……

……

……

……

……

Ne rien écrire ici

3. On se propose de transformer ce principe de recherche en une recherche symétrique par rapport au milieu où on explore, simultanément, les éléments des deux côtés du tableau en se déplaçant vers le centre comme suit :

- 1ère itération : comparer ***a*** avec le premier élément et le dernier élément de ***T***,
- 2ème itération : comparer ***a*** avec le deuxième élément et l'avant dernier élément de ***T***,

  ⋮

- la recherche s'arrête lorsque ***a*** est trouvé ou il n'y a plus d'éléments à comparer avec ***a***.

En se référant à l'algorithme de la fonction **Existe**, initialement donné dans l'énoncé, compléter l'algorithme de la fonction **Existe1** ci-dessous, pour simuler ce nouveau principe de recherche.

Fonction Existe1 (a : Réel, T : Tab, n : Entier) : Entier
DEBUT

……
……
……
……
……
……
……
……
……
……
……
……

FIN

T.D.O.L.

Objet	Type/Nature

Voir suite ☞

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session 2024
	Épreuve : **Informatique**	Sections : **Mathématiques, Sciences expérimentales et Sciences techniques**
	Durée : **1h 30**	Coefficient de l'épreuve : **0.5**

N° d'inscription | | | | | | |

Problème (12 points)

À la fin d'un tournoi, une ligue régionale des amateurs de Basket-ball attribue le trophée **M**ost **V**alue **P**layer (**MVP**) au meilleur joueur ayant eu le plus grand impact sur son équipe.

L'élection du joueur **MVP** est réalisée par un groupe de journalistes sportifs considérés comme des experts. Chaque journaliste vote pour trois joueurs en les classant par ordre de préférence. Selon cet ordre, on accorde **5 points** au premier choix, **3 points** au deuxième choix et **1 point** au troisième choix. À la clôture de l'élection, le joueur ayant obtenu le plus grand nombre de points au total est déclaré comme joueur **MVP**.

Afin d'automatiser l'élection du joueur **MVP**, on se propose d'écrire une solution algorithmique permettant :

- de remplir un tableau **TJ** par **n** noms de joueurs distincts et composés uniquement de lettres alphabétiques et d'espaces (avec $5 \leq \mathbf{n} \leq 30$).
- de déterminer, dans un tableau **TS**, le total des points cumulés des joueurs en fonction des votes des **m** journalistes (avec $2 \leq \mathbf{m} \leq 50$) sachant que :
    - chaque journaliste doit obligatoirement proposer, dans son vote, trois noms de joueurs distincts et existants dans le tableau **TJ** ;
    - **TS[i]** est le total des points cumulés du joueur ayant pour nom **TJ[i]**.
- d'afficher le score **MVP** de la saison, ainsi que le nom de chaque joueur ayant ce score tout en respectant le format suivant :

    Le score MVP est : **valeur**

    Le(s) joueur(s) MVP : **nom1**, **nom2**, **….**

Exemple :

Soit **TJ** un tableau contenant **six** noms de joueurs.

TJ	Ali Saidi	Maher Bahri	Sami Doggi	Rami Daoued	Seif Toumi	Ouday Kefi
	0	1	2	3	4	5

L'élection du joueur MVP, réalisée par **trois** journalistes, se déroule comme suit :

Au départ, le tableau **TS** est initialisé à zéro.

TS	0	0	0	0	0	0
	0	1	2	3	4	5

Puis, après chaque vote, les totaux des points cumulés des trois joueurs choisis sont mis à jour.

- **Vote du 1er Journaliste :**
  Choix 1 : Rami Daoued
  Choix 2 : Ali Saidi
  Choix 3 : Seif Toumi

  ➔ Les éléments TS[3], TS[0] et TS[4] sont incrémentés respectivement de 5, 3 et 1.

TS	**3**	0	0	**5**	**1**	0
	0	1	2	3	4	5

- **Vote du 2ème Journaliste :**
  Choix 1 : Maher Bahri
  Choix 2 : Ali Saidi
  Choix 3 : Rami Daoued

  ➔ Les éléments TS[1], TS[0] et TS[3] sont incrémentés respectivement de 5, 3 et 1.

TS	**6**	**5**	0	**6**	1	0
	0	1	2	3	4	5

- **Vote du 3ème Journaliste :**
  Choix 1 : Ali Saidi
  Choix 2 : Rami Daoued
  Choix 3 : Maher Bahri

  ➔ Les éléments TS[0], TS[3] et TS[1] sont incrémentés respectivement de 5, 3 et 1.

TS	**11**	**6**	0	**9**	1	0
	0	1	2	3	4	5

Le résultat à afficher est :

Le score MVP est : **11**
Le(s) joueur(s) MVP : **Ali Saidi**

Travail demandé :

1. Ecrire un algorithme du programme principal, solution à ce problème, en le décomposant en modules.
2. Ecrire un algorithme pour chaque module envisagé.

N.B. Le candidat est appelé à utiliser les modules ci-dessous sans les développer :

- **Test(ch)** : une fonction booléenne qui retourne **vrai** si la chaîne de caractères **ch** se compose de lettres alphabétiques et d'espaces seulement, ou retourne **faux** dans le cas contraire.
- **Existe(ch, T, n)** : une fonction qui vérifie l'existence d'une chaîne de caractères **ch** dans un tableau **T** de **n** éléments de type chaîne de caractères. Cette fonction retourne soit l'**indice** de la case du tableau **T** contenant **ch**, soit la valeur **-1** si **ch** n'existe pas dans **T**.