Le sujet comporte 4 pages (numérotés de 1 à 4)

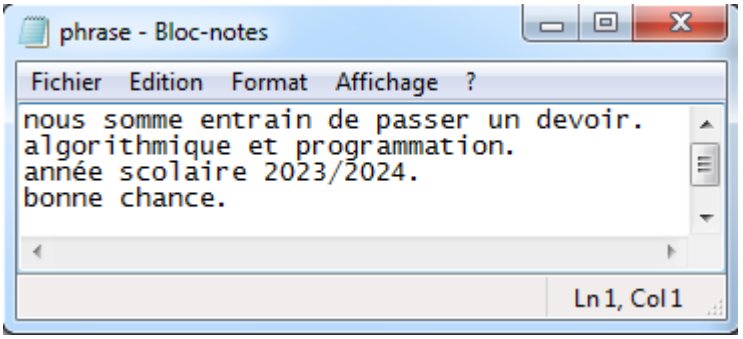
Le sujet comporte 6 pages. les pages 1/6,2/6, et 3/6 seront remise à la fin de l'épreuve

Nom & Prénom ……

EXERCICE 1 (3 Points)

On dispose d'un fichier texte **"phrase.txt"** dont chaque ligne est formée par des mots (on suppose que la ligne contient 15 mots au maximum). Deux mots sont séparés par un seul espace et chaque ligne se termine par un point.

Exemple d'un fichier :



On désire former un fichier binaire **"resultat.dat"** où chaque enregistrement est formé par trois champs :

- **numL** : numéro de la ligne dans le fichier texte précédé par la lettre L.
- **nb_mots** : représente le nombre de mots dans chaque ligne.
- **Tmots** : tableau contenants les nb_mots (chaque mot dans une case)

NB :

Soient les noms physiques des fichiers :

Np1 □ "phrase.txt"

Np2 □ "resultat.dat"

1- Pour chacune des questions suivantes, valider chaque proposition par **V** si la réponse est correcte ou **F** dans le cas contraire.

a) La déclaration de l'enregistrement peut être :

Enr = enregistrement numL : entier nb_mots : entier Tmots : tableau de 15 chaînes Fin	Enr = enregistrement numL : chaîne nb_mots : entier Tmots : tab Fin	Enr = enregistrement numL : chaîne nb_mots : entier Tmots : tableau de 15 chaînes Fin

b) L'accès à une case numéro **i** du tableau **Tmots** d'une variable **e** de type **Enr** se fait de la manière suivante (stocker la chaîne ″devoir″ dans la case numéro i):

e.Tmots □″devoir″	Tmots[i] □ ″devoir″	e.Tmots[i]□ ″devoir″

c) Pour former le fichier **Np2** à partir du fichier **Np1**, on écrit :

```
Procédure former(np2,np1: chaîne)
  Début
     Ouvrir (np1, ft,"r")
     Ouvrir (np2, fb,"wb")
      i □ 0
     Tant que non fin_fichier(ft) faire
        Lire_ligne(ft,ch)
        e.numL □ "L"+convch(i)
        ch □ ch + " "
        e.nb_mots□ 0
        Tant que (ch ≠ "") faire
           p □ pos(" ",ch)
           e.Tmots[e.nb_mots]□ sous_chaîne(ch,0,p)
           e.nb_mots□ e.nb_mots + 1
           ch □ sous_chaîne(ch,p+1,long(ch))
        fin tant que
        Ecrire (fb,e)
        i □ i + 1
     fin tant que
     fermer (ft)
     fermer (fb)
  fin
```

```
Procédure former(np2,np1: chaîne)
Début
  Ouvrir (np1, ft,"r")
  Ouvrir (np2, fb,"wb")
   i □ 0
  Tant que non fin_fichier(ft) faire
     Lire_ligne(ft,ch)
     e.numL □ "L"+convch(i)
     e.nb_mots □ 0
     Tant que (pos(" ",ch) ≠ -1) faire
       p □ pos(" ",ch)
       e.Tmots[e.nb_mots]□ sous_chaîne(ch,0,p)
       e.nb_mots□ e.nb_mots + 1
       ch □ sous_chaîne(ch,p+1,long(ch))
     fin tant que
     e.Tmots[e.nb_mots]□ sous_chaîne(ch,0,p)
     e.nb_mots□ e.nb_mots + 1
     Ecrire (fb,e)
     i □ i + 1
  fin tant que
  fermer (ft)
  fermer (fb)
  fin
```

2- Corriger les erreurs détectées dans cette procédure qui permet d'afficher le fichier ″resultat.dat″:

```
Procédure affiche (Np2 : fich)
   Début
      Ouvrir (np2,fb,"ab")
       Tant que non fin_fichier faire
            Lire_ligne(fb, e)
            Ecrire (e.numL)
            Ecrire (e.nb_mots)
            Ecrire(e.Tmots)
      Fin tant que
   Fermer(fb)
   fin
```

```
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
```

EXERCICE 2 (1.5 Points)

Soit la fonction inconnu suivante :

```
Fonction inconnu (X, B : entier) :…….
    Début
    Ch □ ""
    Tant que non (X = 0) faire
        C □ convch(X mod B)
        X □ X div B
        Si long(C ) > 1 alors
            Ch □ chr(valeur(C) + 55) + Ch
        Sinon
            Ch □ C + Ch
        Fin si
    Fin tant que
    Retourner Ch

    Fin
```

Questions :

- Compléter le type de la fonction.
- Déclarer les objets utilisés dans cette fonction.
- Exécuter manuellement cette fonction et donner les résultats correspondants pour les exemples suivants :
- Déduire le rôle de cette fonction.

……
……
……
……

EXERCICE 3 (3 Points)

Le tri Radix, est un algorithme de tri qui trie les éléments d'un tableau d'entiers positifs en examinant leurs chiffres du moins significatif (chiffre d'unités) au plus significatif en allant de droite à gauche.

On vous donne le principe de fonctionnement de cette méthode de tri :

1. **Initialisation :** commencer par chercher le maximum des éléments pour savoir le nombre de chiffres maximal.
2. **Tri :** Les éléments sont triés selon le chiffre le moins significatif (chiffre des unités). Cette opération est généralement réalisée à l'aide d'un algorithme de tri connu, tel que le tri à bulles.
3. **Répétition :** Répéter l'étape 1 pour chaque chiffre, en se déplaçant du chiffre le moins significatif au chiffre le plus significatif (de droite à gauche).
4. **Résultat final :** Après avoir traité tous les chiffres, le tableau résultant est complètement trié.

Exemple

T	320	459	2	190	957	7384

Étape 1: Chercher le nombre de chiffre **Max = 4**

Étape 2: Trier le tableau selon l'ordre croissant du chiffre d'unités.

T	32**0**	19**0**	**2**	738**4**	95**7**	45**9**

Trier le tableau selon l'ordre croissant des chiffres de dizaines.

T						
	2	320	957	459	7384	190

Trier le tableau selon l'ordre croissant des chiffres de centaines.

T						
	2	190	320	7384	459	957

Trier le tableau selon l'ordre croissant des chiffres de milliers.

T						
	2	190	320	459	957	7384

On vous donne l'algorithme suivant :

```
Procédure  TriRadix  ( @T:Tab, N:entier)
DÉBUT
        Maxi ← NbChiffreMax (T ,N)        # Trouver le nombre de chiffres le plus élevé dans le tableau
        # Effectuer le tri croissant selon chaque chiffre, du moins significatif au plus significatif
        Pour x de  1  à Maxi  faire
            Tri_tableau (……..)
        Fin pour
FIN
```

Travail demandé

1) Donner le tableau de déclarations des objets locaux

2) Donner les algorithmes des modules **NbChiffresMax** et **Tri_tableau** (en précisant ses paramètres) utilisés par la procédure **TriRadix**

EXERCICE 4 (4.5 Points)

On part d'un entier n auquel on fait subir une transformation :

- ✔ Si n est pair, on le devise par deux.
- ✔ Si n est impair, on le multiplie par 3 et on ajoute 1.
- ✔ Puis, on recommence sur le résultat.

NB : quelque soit l'entier n, on finit par retomber sur 1

Par exemple, en partant de n=10, on obtient :

10 ; 5 ; 16 ; 8 ; 4 ; 2 ; 1 ;

1- Déclarer la carte d'identité d'un entier comme l'enregistrement de :

- cet entier,
- sa trajectoire (les entiers rencontrés jusqu'à 1),
- sa durée de vol (le nombre d'entiers rencontrés avant de trouver 1),

Valeur initiale = 10

Trajectoire ="10 ; 5 ; 16 ; 8 ; 4 ; 2 ; 1"

Durée de vol = 6

Altitude maximale = 16

- son altitude maximale (le plus grand entier rencontré).

2- Ecrire l'algorithme d'un module qui permet de remplir le fichier ″carte.dat″ par NB cartes d'identités **d'entiers aléatoires distincts de deux chiffres** dans un fichier ″carte.dat″, sachant que NB est un entier déjà saisi dans le programme principal.

EXERCICE 5 (8 Points)

Pour protéger l'accès à votre mobile, on peut utiliser un mot de passe ou un schéma de déverrouillage.



La deuxième méthode est la plus utilisé par les utilisateurs des smartphones.

Dans ce sens on veut concevoir un programme qui permet de :

1. Lire un schéma de déverrouillage.
2. Générer un mot de passe correspondant au schéma utilisé en cas où l'utilisateur oublie son schéma de déverrouillage.
3. Générer une clé d'authentification donnée à l'utilisateur lors de la création de son schéma, et demandé par le système de sécurité lorsque l'utilisateur oublie son schéma pour lui fournir le mot de passe généré dans l'étape **2.** .

Le programme affiche à l'utilisateur une matrice **carré** de taille **3** pour choisir son schéma.

La matrice est composée des enregistrements où chaque enregistrement contient **3 champs** :

- **Car** : contient une lettre ou un chiffre. Ce champ doit être rempli d'une manière aléatoire.

  (ord ("0") = 48 , ord("A") = 65 , ord ("a") = 97)

- **Val** : doit être initialisé par le caractère '*', puis et après le choix du schéma peut contenir le caractère :
  - **'S'** : si la case fait partie du schéma de déverrouillage.
  - **'D'** : si la case est le début du schéma de déverrouillage.
  - **'F'** : si la case est la fin du schéma de déverrouillage.



- **Dir :** direction de la case suivante, doit être initialisé par le caractère '**-**', puis et après le choix du schéma peut contenir le caractère : (**H**(Haut), **B**(bas), **Dr**(Droite), **G**(Gauche))

  Pour simplifier le traitement, on suppose que les diagonales ne feront pas partie du schéma

Question1 : Déclarer un type pour l'enregistrement et un type pour la matrice

Question2 : Ecrire l'algorithme d'un module qui permet d'initialiser cette matrice comme indiqué ci-dessous.

Question3 : Ecrire l'algorithme du module qui permet de modifier le contenu de la matrice en fonction du schéma donné par l'utilisateur sachant que le schéma est une chaine formée par les indices des cases séparées par un tiret

(Le candidat n'est pas appelé à saisir cette chaine)

Si la chaine est 20-21-11-01-00

20 désigne ligne 2 colonne 0

21 désigne ligne 2 colonne 1

11 désigne ligne 1 colonne 1

01 désigne ligne 0 colonne 1

00 désigne ligne 0 colonne 0

Les cases de la matrice qui seront modifiés :

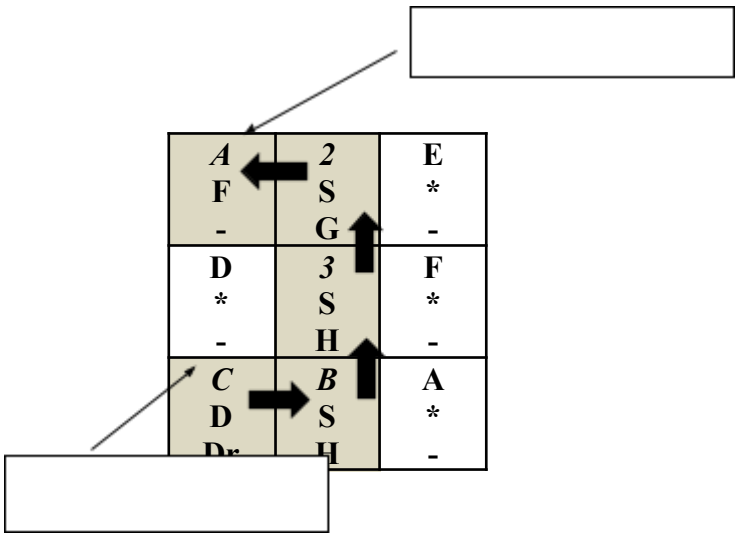
- M[2,0] : contient dans le champ **val** la lettre **D** et dans le champ **Dir** la valeur **Dr** puisque l'indice de la colonne de la case suivante est incrémenté de 1 (2**0**-21 **0+1 =1)**
- M[2,1] : contient dans le champ **val** la lettre **S** et dans le champ **Dir** la valeur **H** puisque l'indice de la ligne de la case suivante est décrémenté de 1 (**2**1-**1**1**)**
- M[1,1] : contient dans le champ **val** la lettre **S** et dans le champ **Dir** la valeur **H** puisque l'indice de la ligne de la case suivante est décrémenté de 1 (**1**1-**0**1**)**
- M[0,1] : contient dans le champ **val** la lettre **S** et dans le champ **Dir** la valeur **G** puisque l'indice de la colonne de la case suivante est décrémenté de 1 (0**1**-0**0)**
- M[0,0] : contient dans le champ **val** la lettre **F** puisque c'est la dernière case du schéma

Question 4 : Ecrire l'algorithme d'un module qui permet à partir de la chaine ch donnée est de la matrice M modifiée de générer le mot de passe correspond et sa clé d'authentification sachant que :

- Le mot de passe est la concaténation des caractères des cases de la matrice qui forment le schéma.
- La clé d'authentification est la concaténation des codes ASCII des caractères du mot de passe.

Exemple :

Si ch = 20-21-11-01-00 et la matrice après le choix du schéma de déverrouillage est :



Alors le mot de passe qui sera générer par le programme est : **CB32A**

Et La clé est : **6766515065**