

## Devoir de Contrôle n°2 (TP)

- 1) Une solution modulaire au problème est exigée.
- 2) Enregistrer au fur et à mesure votre code sur le dossier « C:\3SI02\NomEleve »

**Enoncé :**

Le **CAPTCHA** est un code permettant de différencier de manière automatisée un utilisateur humain d'un ordinateur. Ce test est utilisé sur Internet pour vérifier que l'utilisateur n'est pas un robot.

On vous propose d'écrire un programme en pascal permettant de générer des codes CAPTCHA et de les afficher dans un ordre précis, pour cela :

- Remplir un tableau T par N codes CAPTCHA **distincts** sachant qu'un CAPTCHA est une chaîne de **cing** caractères **alphanumériques** ['A'..'Z', '0'..'9'] et ( $5 \leq N \leq 20$ ).
- Afficher le tableau T avant le tri.
- Trier les éléments de ce tableau dans l'ordre **décroissant** selon le principe suivant :
  - ⇒ Dans un deuxième tableau T1 de N entiers calculer la somme des codes ASCII des caractères de chaque élément de T.
  - ⇒ Pour chaque élément de T1 :
    - Déterminer le nombre (NB) d'éléments qui lui sont supérieurs.
    - Placer cet élément dans un tableau R1 à la case d'indice (NB + 1) et
    - Effectuer l'action nécessaire pour placer l'élément correspondant de T dans sa bonne position en se basant sur le tableau R1
- Afficher le tableau trié.

**Exemple** : Pour n = 7

|     |         |         |         |         |         |         |         |
|-----|---------|---------|---------|---------|---------|---------|---------|
| T : | 'EJ36S' | 'G1M51' | '29KLP' | 'V2V0Z' | 'Q891A' | '33DXU' | 'NC17B' |
|     | 1       | 2       | 3       | 4       | 5       | 6       | 7       |

On obtient alors le tableau T1 suivant (chaque élément T1[i] représente la somme des codes ASCII des caractères de chaque T[i])

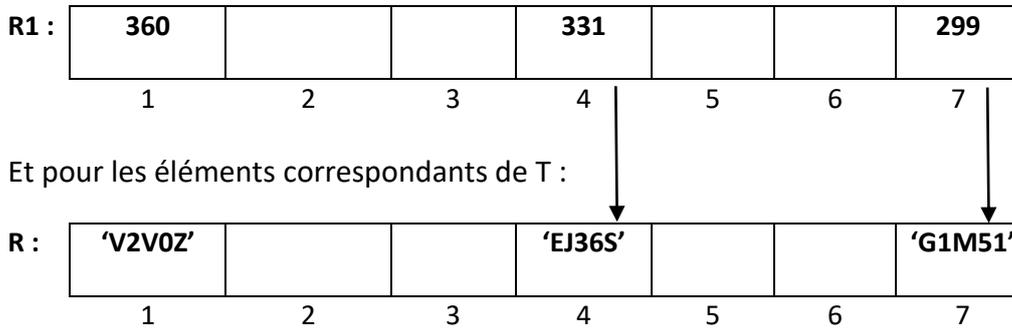
|      |     |     |     |     |     |     |     |
|------|-----|-----|-----|-----|-----|-----|-----|
| T 1: | 331 | 299 | 338 | 360 | 308 | 343 | 315 |
|      | 1   | 2   | 3   | 4   | 5   | 6   | 7   |

Pour l'élément T1[1] = 331, il y'a 3 éléments qui lui sont supérieurs (NB = 3) alors il sera placé à la case d'indice (NB + 1 = 4) dans le tableau R1.

Pour l'élément T1[2] = 299, il y'a 6 éléments qui lui sont supérieurs (NB = 6) alors il sera placé à la case d'indice (NB + 1 = 7) dans le tableau R1.

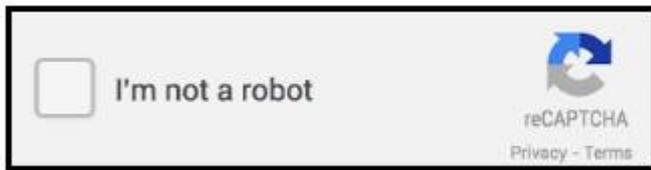
.....

Pour l'élément  $T1[4] = 360$ , il n'y a aucun élément qui lui est supérieur ( $NB = 0$ ) alors il sera placé à la case d'indice ( $NB + 1 = 1$ ) dans le tableau  $R1$ .



En fin le programme affiche le tableau trié.

Voici un exemple de CAPTCHA :



Grille d'évaluation :

| Actions                               | Nombre de points |
|---------------------------------------|------------------|
| <i>Si programme exécutable Alors</i>  | <b>20</b>        |
| <i>Sinon</i>                          |                  |
| - Décomposition et appels des modules |                  |
| - Compilation des erreurs             |                  |
| - Développement des modules :         |                  |
| • Saisir la taille du tableau         |                  |
| • Remplir le tableau T                |                  |
| • Affichage du tableau T              |                  |
| • Calcul des sommes dans T1           |                  |
| • Trier R1 et donc R                  |                  |
| • Affichage du tableau trié           |                  |

Bon Travail 😊

```

Program DC2_TP;
Uses Wincrt;
Type
  tab = Array[1..30] Of String;
  tab1 = Array[1..30] Of Integer ;
Var
  t,r: tab ;
  n : Integer;
  t1,r1 : tab1;
{**** Procédure saisir ****}

Procedure saisir(Var n : Integer);
Begin
  Writeln('Taper le nombre des éléments de T : ');
  Repeat
    Write('n= ');
    read(n);
  Until (n>=5) And ( n <= 20);
End;
{*****@KiraiEEE*****}
Function verif(ch:String): Boolean;
Var
  i: Integer;
  test: Boolean;
Begin
  i := 0;
  Repeat
    i := i+1;
    If ch[i] In ['A'..'Z','0'..'9'] Then test := True
    Else test := False;
  Until (Not(test)) Or (i>=Length(ch));
  verif := (test And (Length(ch)=5));
End;
{*****@KiraiEEE*****}
Function distinct(n: Integer; t: tab): Boolean;
Var
  exist: Boolean;
  j: Integer;
Begin

```



```

j := 0;
exist := False;
Repeat
    j := j + 1;
    exist := (t[j]=t[n]);
Until ( exist) Or( j=n-1);
distinct := exist;
End;
Procedure remplir(n: Integer; Var t : tab );
Var
    i : Integer;
Begin
    Writeln;
    Writeln('** Taper les elements de T : **');
    Repeat
        Write('t[1] = ');
        read(t[1]);
    Until (verif(t[1]));
    For i:=2 To n Do
        Begin
            Repeat
                Write('t[' ,i, ' ] = ');
                read(t[i]);
            Until (verif(t[i]) ) And (distinct(i,t));
        End;
    Writeln;
End;
Procedure afficher(n: Integer; Var t : tab );
Var
    i : Integer;
Begin
    For i := 1 To n Do
        Begin
            Write(t[i], ' | ');
        End;
    Writeln;
End;
{**** somme des codes ASCII des caractères d'une chaine ****}
Function somme(ch: String): Integer;

```



```

Var
  s,j : Integer;
Begin
  s := 0;
  For j:=1 To Length(ch) Do
    Begin
      s := s+ Ord(ch[j]);
    End;
  somme := s;
End;
{**** remplissage de T1 ****}
Procedure remp(n: Integer; t: tab; Var t1: tab1);
Var
  i: Integer ;
Begin
  For i:= 1 To n Do
    Begin
      t1[i] := somme(t[i]);
    End;
  End;
{**** affichage du tableau des sommes ****}
Procedure affi(n: Integer; t1: tab1);
Var
  i: Integer;
Begin
  Writeln;
  For i:= 1 To n Do
    Begin
      Write(t1[i], ' | ');
    End;
  End;
Procedure trier(n: Integer; Var t1,r1: tab1; Var t,r: tab);
Var
  i,j,nb: Integer;
Begin
  For i:=1 To n Do
    Begin
      nb := 0;

```



```

For j:=1 To n Do
  Begin
    If t1[i]< t1[j] Then
      nb := nb + 1;
    End;
    r1[nb+1] := t1[i];
    r[nb+1] := t[i];
  End;
End;
Begin
  Writeln;
  saisir(n);
  remplir(n,t);
  Writeln(' ---- Tableau T initial : -----');
  afficher(n,t);
  remp(n,t,t1);
  affi(n,t1);
  trier(n,t1,r1,t,r);
  {*** Affichage du tableau trié ***}
  Writeln(' ---- Tableau T Trié : -----');
  afficher(n,r);
End.

```

