NF01 – Printemps 2019

Examen Final

ATTENTION !
Utilisez quatre copies séparées, une par partie.

Document autorisé : une feuille de notes A4 recto-verso manuscrite.

1ère Partie : Championnat d'échecs *(5 points)*

Vous êtes un des superviseurs du championnat d'échecs qui se tiendra ce week-end. On vous demande d'écrire un programme en Pascal, afin de mieux gérer les résultats. Ce programme doit prendre en compte les informations suivantes pour chaque joueur : son nom (`nom`), le nombre de parties qu'il a jouées (`nbParties`), le nombre de points total qu'il a obtenus (`score`) , l'écart de ce nombre par rapport à la moyenne des points de tous les joueurs (`ecart`) . Un joueur obtient 2 points s'il gagne une partie, aucun point s'il la perd, 1 point en cas de partie nulle. Le score est le cumul des points obtenus.

a. Définir le type correspondant à un joueur ainsi que le type tableau permettant de gérer l'ensemble des joueurs.

b. Écrire une procédure `saisie` de saisie des informations de l'ensemble des joueurs, avec deux arguments : le tableau de joueurs et sa taille.

c. Écrire une procédure qui calcule pour chaque joueur l'écart de son score par rapport à la moyenne des scores.

   N.B. : On définira une fonction permettant de calculer cette moyenne.

d. Écrire une procédure permettant de trier par ordre décroissant des scores le tableau des joueurs.

e. Afficher le nom et le score des joueurs qui ont obtenu un score supérieur à la moyenne, ainsi que le nombre de joueurs correspondant.

```
program Championnat;
const
   NMAX=100 ;
type
   Joueur = record
                  nom : string ;
                  score : integer;
                  nbParties : integer;
                  ecart: real;
            end;

   TJoueurs = array[1..NMAX] of Joueur;

var
   size : integer;
   t : TJoueurs;

procedure saisie (var tab : TJoueurs ; taille : integer);
var
   i: integer;
```

```
begin
      for i:=1 to taille do
      begin
           write('nom du joueur ? ', i);
           readln(tab[i].nom);
           write('score ? ');
           readln(tab[i].score);
           write('Nombre de parties ? ');
           readln(tab[i].nbParties);
      end;
end;

function moyenne(tab : TJoueurs; taille :integer) : real;
var
   i,sum: integer;
begin
      sum := 0;
      for i:=1 to taille do
      begin
           sum := sum + tab[i].score;
      end;
      moyenne := sum/i;
end;

procedure remplir_ecart (var tab:TJoueurs ;taille:integer);
var
   i: integer;
begin
      for i:=1 to taille do
      begin
           tab[i].ecart:= tab[i].score - moyenne(tab,taille);
      end;
end;

procedure trier (var t : TJoueurs; taille: integer);
var
   i,j: integer;
   temp : Joueur;
begin
     for i:=1 to taille do
      for j:=i+1 to taille do
          if t[i].ecart < t[j].ecart then
            begin
              temp:=t[i];
              t[i]:=t[j];
              t[j]:=temp;
            end;
end;

procedure afficherMeilleurs(tab :TJoueurs; taille:integer);
var
   i,j: integer;
begin
      trier(t,taille)
      j := 0;
      for i:=1 to taille do
      begin
           if tab[i].ecart > 0 then
           begin
              write('nom : ',tab[i].nom);
              write('score : ', tab[i].score);
              j := j + 1;
```

```
            end;
        end;
        writeln(j,' joueurs ont un score supérieur à la moyenne') ;
end;

begin
          repeat
             write('Nb de joueurs');
             readln(size);
          until size <= NMAX
          saisie (t, size);
          writeln('moyenne : ',moyenne(t,size));
          trier (t,size);
          afficherMeilleurs(t,size);

end.
```

{ ---------- pensez à changer de copie ---------- }

2ème Partie : Nombres à ne pas prononcer *(5 points)*

Un nombre imprononçable (unspeakable number) est un nombre divisible par 7, ou qui contient le chiffre 7 dans les chiffres qui le composent.

Exemples : 7 ; 14 ; 17 ; 21 ; 27 ; 1072 ; 945 ; - 570

Le jeu des nombres imprononçables consiste à essayer de compter rapidement aussi loin que l'on peut en sautant les nombres imprononçables. On génère la série suivante notée (*) :

(*) : 1 2 3 4 5 6 ( ) 8 9 10 11 12 13 ( ) 15 16 ( ) 18 19 20 ( ) 22 23 24 25 26 ( ) ( ) 29 30…

1) Ecrire la fonction **récursive** `contient_7(n : integer): boolean` qui détermine si un entier signé `n` contient le chiffre 7.
   Exemples :
   `contient_7(575)` retourne `true`
   `contient_7(-9)` retourne `false`

2) Ecrire la procédure **récursive** `generer(NMAX : integer)` qui génère la série du jeu des nombre imprononçables (*) entre les bornes `-NMAX` et `NMAX`.

```
function contient_7(N : integer):Boolean;
begin
      //Condition d'arrêt 1 : trouver un 7 dans les unités
      if (N mod 10 = 7) then
            contient_7 := true
      else
            //Condition d'arrêt 2 : on est arrivés au 0
            if N = 0 then
                  contient_7 := false
            else
                  if N < 0 then
                        //appel récursif 1 (entiers négatifs)
                        contient_7 := contient_7(-N)
                  else
                        //appel récursif 2 (chiffre suivant)
                        contient_7 := contient_7(N div 10);
end ;

procedure generer(NMAX : integer) ;
begin
      if (NMAX = 0) then
         write(NMAX)
      else
      begin
            if  not (contient_7(NMAX) or ((NMAX mod 7) = 0)) then
            begin
                  write(-NMAX,' ');
                  generer(NMAX-1);
                  write(' ',NMAX);
            end
            else
                generer(NMAX-1);
      end;
end;
```

{ ---------- pensez à changer de copie ---------- }

3ème Partie : Chaines de caractères *(5 points)*

Exercice 1 : Fonction premier mot *(1,5 points)*

Écrire en Pascal une fonction `premierMot(chaine)` qui renvoie le premier mot d'une chaîne de caractères. Par exemple si ma chaîne est «samedi soir, je vais au cinéma», la fonction renverra «samedi».

N.B. : une chaine peut commencer par des caractères de ponctuation ou des espaces. Ainsi le premier mot de la chaine «… samedi soir, je vais au cinéma» est samedi.

Exercice 2 : Codes *(1 point)*

Écrire en Pascal une fonction qui affiche les caractères d'une chaine dont le code ASCII est compris entre les nombres 105 et 110.

Exercice 3 : Codage de César *(1 point)*

Écrire en Pascal une fonction codage qui prend en argument une chaîne de caractères et décale ses lettres de trois crans dans l'alphabet. Par exemple, «bac» sera transformée en «edf».

Écrire en Pascal la fonction de décodage.

Exercice 4 : Inverse (1.5 points)

Écrire en Pascal une procédure `inverse(var f :TEXT)` qui prend en argument un fichier texte et qui inverse toutes les lettres du fichier.

Par exemple, si fichier f contient la chaîne 'Samedi soir je vais au cinéma', inverse(f) retournera un fichier contenant 'aménic ua siav ej rios idemaS'.

```
Program ChaineDeCaractere;
var
  s : string;
  n, choix : integer;
  f : TEXT;

function premierMot (s : string) : string;
var
  i, j : integer;

begin
    j := 1;
    // Partie si la phrase ne commence pas par une lettre
    while (not ((s[j] in ['a'..'z']) OR (s[j] in ['A'..'Z']))) do
        j := j + 1;
    // Fin partie si la phrase ne commence par une lettre

    i := j;
    while (s[i] in ['a'..'z']) OR (s[i] in ['A'..'Z']) do
        i := i + 1;
   premierMot := copy(s, j, i-j);
   writeln('Premier mot : ', premierMot);
end;

procedure ascii (s : string);
var
  i : integer;

begin
    for i := 1 to length(s) do
       if ord(s[i]) in [105..110] then
          write(s[i]);
    writeln;
end;

procedure codageCesar (var s : string);
var
  i : integer;

begin
    for i := 1 to length(s) do
        s[i] := chr(ord(s[i])+3);

end;
```

```
procedure deCodageCesar (var s : string);
var
   i : integer;

begin
     for i := 1 to length(s) do
          s[i] := chr(ord(s[i])-3);

end;

procedure ecriture (var f : TEXT);
var
   s : string;
begin
     writeln('Saisir le fichier');
     readln(s);
     write(f, s);
end;


procedure inverse(var f : TEXT);
var
   s : string;
   c : char;
begin
     reset(f);
     s := '';

     while not eof(f) do
          begin
               read(f, c);
               s := c + s;
          end;
     rewrite(f);

     write(f, s);
end;

procedure lecture (var f : TEXT);
var
   s : string;
begin
    reset(f);
    read(f, s);
    writeln(s);
end;
BEGIN
{ programme non demandé }
     repeat
       repeat
          writeln;
          writeln('================= Menu ================');
          writeln('1-Fonction 1er Mot');
          writeln('2-Fonction ASCII');
          writeln('3-Fonction Codage Cesar');
```

```
        writeln('4-Fonction deodage Cesar');
        writeln('5-Fonction Inverse');
        writeln('0-sortir du menu');
        writeln('=======================================');

        writeln('Entrer votre choix');
        readln(choix);
    until choix in [0..5];


    case choix of
        1: begin
              writeln('Saisir une phrase :'); readln(s);
              s := premierMot(s);
              writeln('Le premier mot est : ', s);
           end;
        2: begin
              writeln('Saisir une phrase :'); readln(s);
              ascii(s);
           end;
        3: begin
              writeln('Saisir une phrase :'); readln(s);
              codageCesar(s);
              writeln('Codage : ', s);
           end;
        4: begin
              deCodageCesar(s);
              writeln('Decodage : ', s);
           end;
         5: begin
              assign(f, 'inverse');
              rewrite(f);
              ecriture(f);
              inverse(f);
              lecture(f);
           end;

        end;


   Until choix=0;

   readln;
end.
```

{ ---------- pensez à changer de copie ---------- }

4ème Partie : Randonnée (5 points)

Vous partez bientôt en randonnée avec des amis. Afin contrôler le remplissage et le contenu de vos sacs à dos, vous décidez de développer un programme en Pascal.

1. Les objets que les randonneurs peuvent mettre dans les sacs à dos sont : sacCouchage, chaussures, chaussettes, linge, gourde, pain. Donnez la définition d'un type enregistrement `Objets` qui

contient l'information sur la présence ou non (`O/N`) de ces objets dans un sac ainsi que le numéro du sac.

2. Écrire la procédure à un argument `remplirSac` qui permet de déposer des objets (parmi ceux définis dans `Objets`) dans un sac. Cette procédure est appelée avec un sac vide et remplit ce sac avec les objets déposés.
3. Écrire la procédure `gestionSacs` qui permet le stockage des sacs remplis dans un fichier de sacs. Le nombre des randonneurs n'est pas connu.
4. Écrire la procédure `afficheSac` qui affiche à l'écran les objets présents dans un sac à dos.
5. Écrire la procédure `afficherSac` qui permet d'afficher le contenu des sacs à dos stockés dans votre fichier.
6. Écrire le programme principal permettant de constituer le fichier de sacs, puis d'afficher le contenu des sacs.

```
program Final;
type
   Objets = record
               sacCouchage, chaussures, chaussettes,
               linge, gourde, pain : boolean;
               num : integer;
            end;

            Fichier = file of Objets;

 var
            fich : Fichier;


 procedure remplirSac (var sac : Objets);
 begin
          with sac do
               begin
                    writeln('Avez vous des chaussures (O/N) ?');
                    readln(chaussures);
                    writeln('Avez vous des chaussettes (O/N) ?');
                    readln(chaussettes);
                    writeln('Avez vous du linge (O/N) ?');
                    readln(linge);
                    writeln('Avez vous une gourde (O/N) ?');
                    readln(gourde);
                    writeln('Avez vous du pain (O/N) ?');
                    readln(pain);
                    writeln('N° randonneur');
                    readln(num);
               end;
 end;

procedure gestionSacs(var f:fichier);
var   rep : char;
      temp : Objets;

begin
      repeat
            remplirSac(temp);
            write(f,temp);
```

```
            writeln('Continuer O/N');
            read(rep);
       until (rep = 'N');
end;

procedure afficheSac(s:Objets);
begin
     with s do
               write(chaussures, chaussettes, linge, gourde, pain, Nb);
     readln;
end;

procedure afficherSac(var f:fichier);
var temp : Objets;
begin
     while not(eof(f))do
           begin
                read(f,temp);
                afficheSac(temp);
           end;
      readln;
end;

 begin
       assign(fich, 'final2019');
       rewrite(fich);
       gestionSacs(fich);
       close(fich);
       reset(fich);
       afficherSac(fich);
       close(fich);
end.
```