

NF01 – Printemps 2019

Examen Final

ATTENTION !

Utilisez quatre copies séparées, une par partie.

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

1^{ère} 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.

- Définir le type correspondant à un joueur ainsi que le type tableau permettant de gérer l'ensemble des joueurs.
- Écrire une procédure `saisie` de saisie des informations de l'ensemble des joueurs, avec deux arguments : le tableau de joueurs et sa taille.
- 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.

- Écrire une procédure permettant de trier par ordre décroissant des scores le tableau des joueurs.
- Afficher le nom et le score des joueurs qui ont obtenu un score supérieur à la moyenne, ainsi que le nombre de joueurs correspondant.

f.

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

2^{ème} 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`.

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

3^{ème} 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 chaîne peut commencer par des caractères de ponctuation ou des espaces. Ainsi le premier mot de la chaîne «... samedi soir, je vais au cinéma» est samedi.

Exercice 2 : Codes (1 point)

Écrire en Pascal une fonction qui affiche les caractères d'une chaîne 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'.

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

4^{ème} 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.