

# Examen d'informatique 2002-2003

DEUG 1<sup>ère</sup> année (MASS - MIAS - SM) - Deuxième session - Septembre 2003

Polycopiés et notes de cours autorisés - Calculatrice non autorisée

Durée : 3 heures

## ARCHITECTURE DES ORDINATEURS (coefficient 2)

- a) Est-il possible de coder le nombre entier décimal 263 en binaire sur un octet ? Justifiez votre réponse.
- b) Donnez une valeur approchée en Ko de l'espace mémoire nécessaire pour stocker une image de résolution  $800 \times 500$  pixels en supposant que chaque pixel est codé sur 32 bits et qu'une compression du fichier avec un taux égal à 10 a été réalisée.

## NUMERATION (coefficient 4)

- a) Convertir le nombre binaire  $(1100101, 11)_2$  en décimal, octal et hexadécimal.
- b) Convertir le nombre hexadécimal  $(23F)_{16}$  en base 4.
- c) En codage par complément à 2 (complément à la base) sur 8 bits, le nombre décimal  $x$  est représenté par le code 11100100. Quel est le signe du nombre décimal  $x$  ? Quelle est la valeur absolue du nombre décimal  $x$  ?
- d) Effectuez la soustraction suivante en base octale :  $(7432)_8 - (5567)_8$ .

## COMPREHENSION (coefficient 2)

Vous allez exécuter "à la main" le programme ci-dessous en supposant que la valeur affectée à  $n$  est égale à 8. Quelles sont alors les valeurs affichées par ce programme ?

```
program comprehension ;
var i , n : integer ;
function test(var x : integer ; var y : integer) : boolean ;
begin
  if (x mod y <> 0) then test := false
  else test := true ;
end ;
begin
  readln(n) ;
  for i := 0 to 50 do
    if test(i,n) then writeln(i)
  end.
end.
```

## PROGRAMMATION (coefficient 2)

Il vous est demandé de concevoir un programme qui écrit dans un fichier les valeurs des N premiers éléments (N étant lu au clavier) de la suite récurrente définie comme suit :

$$\begin{aligned}u_0 &= 1 ; \\u_1 &= 2 ; \\u_{i+1} &= u_i + \frac{1}{u_{i-1}} .\end{aligned}$$

Vous déclarerez pour cela une variable **f** de type **text** et lui assignerez le nom de fichier **suite.dat**.

## PROGRAMMATION (coefficient 3)

Ecrire un programme qui permet de lire une série de N mots par l'intermédiaire du clavier, puis qui les affiche par longueur croissante (longueur = nombre de lettres dans le mot). Par exemple, après avoir saisi au clavier les mots suivants :

```
barcelone
paris
rome
londres
```

le programme affichera :

```
rome
paris
londres
barcelone
```

Les tâches à programmer sont les suivantes :

- saisie du nombre N au clavier ;
- saisie des N mots et stockage des N mots dans un tableau T ;
- classement du contenu du tableau T par longueur croissante. Vous utiliserez pour cela un algorithme de tri de votre choix ;
- affichage des mots classés par longueur croissante.

## PROBLEME : programmer un jeu de dé (coefficient 7)

Vous allez programmer un jeu de dé pour deux joueurs dont le principe est le suivant :

- Le premier joueur lance le dé. S'il obtient la face 1, il ne marque aucun point et passe le dé au second joueur. Tant qu'il obtient une face différente du 1, il peut relancer le dé et il cumule les points obtenus à chaque lancer de dé. Il peut décider d'arrêter à tout moment afin de conserver son score et de passer la main au second joueur. Attention : si le joueur obtient la face 1 avant de décider d'arrêter sa série de lancers, il perd la main et le score de cette série de tirage sera de zéro.
- Les deux joueurs jouent à tour de rôle selon le principe décrit ci-dessus. Le gagnant est celui qui obtient le premier un total de 100 points.

Exemple de début de partie :

- Tirage premier joueur : 4, 2, 5 ; total des points = 11
- Tirage second joueur : 3, 1 ; total des points = 0
- Tirage premier joueur : 6, 3 ; total des points = 11 + 9 = 20
- Tirage second joueur : 2, 3, 2 ; total des points = 0 + 7 = 7
- Tirage premier joueur : 5, 4, 1 ; total des points = 20 + 0 = 20
- Tirage second joueur : 2, 4, 6, 5 ; total des points = 7 + 17 = 24
- etc...

Le premier joueur sera l'utilisateur du programme. Il faudra lui proposer à chaque étape la possibilité de relancer le dé ou bien d'arrêter sa série de lancers et de passer la main au second joueur.

Le second joueur sera simulé par la machine. La stratégie que vous allez programmer pour son jeu est la suivante : il réalisera 5 tirages avant de repasser la main au premier joueur. S'il obtient la face 1 au cours de ces tirages, alors il repassera la main avant d'avoir pu réaliser les 5 tirages.

Le programme que vous allez écrire doit être organisé de la façon suivante :

- Déclaration d'une procédure **jeu\_joueur** ayant comme argument (susceptible d'être modifié) le nombre de points **pt\_j** du premier joueur (l'utilisateur du programme). Dans cette procédure, les tâches suivantes devront être programmées :
  - réalisation d'une série de lancers de dé jusqu'à ce que la face obtenue soit de 1 ou bien que le joueur décide d'arrêter la série. Il faut donc lui proposer, après chaque tirage (s'il est différent de 1), la possibilité de continuer la série de lancers ou bien de l'arrêter ;
  - calcul et affichage du score de la série : il est égal à zéro si la face 1 est sortie, sinon, il est égal à la somme des valeurs obtenues avec le dé ;
  - calcul et affichage du nombre de points **pt\_j** : il est égal à la valeur **pt\_j** précédente plus le score obtenu lors de cette série.
- Déclaration d'une procédure **jeu\_machine** ayant comme argument (susceptible d'être modifié) le nombre de points **pt\_m** du second joueur (simulé par la machine). Dans cette procédure, les tâches suivantes devront être programmées :
  - réalisation d'une série de tirage du dé jusqu'à ce que la face obtenue soit de 1 ou bien que le nombre de tirages de cette série s'élève à 5 (en accord avec la stratégie de jeu choisie pour le second joueur) ;
  - calcul et affichage du score de la série : il est égal à zéro si la face 1 est sortie, sinon, il est égal à la somme des valeurs obtenues avec le dé ;
  - calcul et affichage du nombre de points **pt\_m** : il est égal à la valeur **pt\_m** précédente plus le score obtenu lors de cette série.
- Ecriture du programme principal effectuant les tâches suivantes :
  - appels successifs des procédures **jeu\_joueur** et **jeu\_machine** jusqu'à ce que l'un des deux totaux de points, **pt\_j** ou bien **pt\_m**, ait atteint la valeur 100 ;
  - affichage du résultat du jeu : le gagnant et les scores finaux des deux joueurs.

Aide : pour obtenir aléatoirement les valeurs des six faces du dé, vous utiliserez la fonction **random** du langage PASCAL.