Examen d'informatique 2003-2004

DEUG 1^{ere} année (MASS - MIAS - SM) - Première session - Janvier 2004

Polycopié et notes de cours autorisés - Calculatrice non autorisée

Durée : 3 heures

ARCHITECTURE DES ORDINATEURS (coefficient 2, durée ≈ 10 min)

a) A quoi correspondent les unités Ko, Mo et Go en informatique ? Pour quelle raison les utilise-t-on ?

b) Une disquette a pour capacité 1,4 Mo. Si l'on suppose que chaque pixel est codé sur 32 bits, combien d'images de résolution 200×500 pixels peut-on stocker sur cette disquette ?
Combien d'images de résolution similaire peut-on stocker sur un CD-Rom dont la capacité est de l'ordre de 600 Mo ?

NUMERATION (coefficient 4, durée ≈ 40 min)

a) Soit le nombre décimal $(205)_{10}$. Combien de bits sont nécessaires pour coder ce nombre décimal en binaire ? Donnez la représentation de ce nombre dans les bases octale et hexadécimale.

b) Convertir le nombre décimal $(323,3)_{10}$ en base 3. Si la partie fractionnaire possède un développement infini périodique, indiquez sa période.

c) Effectuez directement en base 4 la soustraction $(3202)_4 - (2113)_4$.

d) Soit le code 11011100. Quel nombre décimal représente ce code si l'on suppose avoir utilisé un codage en complément à la base sur 8 bits ? Même question pour un codage en complément restreint sur 8 bits, puis pour un codage en valeur absolue plus signe sur 8 bits.

PROGRAMMATION (coefficient 2, durée ≈ 15 min)

Soient les sommes S_1 et S_2 définies de la façon suivante :

$$\begin{aligned} S_1 &= \sum_{i=1}^{N} i = 1 + 2 + \cdots + N \\ S_2 &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ij \\ &= 1 \times (2 + 3 + \cdots + N) + 2 \times (3 + 4 + \cdots + N) + \cdots + (N-1) \times N \end{aligned}$$

L'entier N étant lu au clavier, écrire un programme en Pascal qui calcule S_1 et S_2.

COMPREHENSION (coefficient 2, durée ≈ 15 min)

Imaginez que vous exécutiez le programme ci-dessous. Quelles seront les deux valeurs affichées ?

```
program comprehension ;
var  x,y:integer ;
procedure b_for(var a:integer) ;
var i:integer ;
begin
     for i := 4 downto 2 do
         a := a*i
end;
function b_repeat(a:integer) : integer ;
var i:integer ;
begin
     i := 2 ;
     repeat
           a := a*i ;
           i := i+1
     until i > 5 ;
     b_repeat := a
end ;
begin
     x := 3 ;
     b_for(x) ;
     writeln(x) ;
     y := 3 ;
     writeln(b_repeat(y))
end.
```

PROGRAMMATION : analyse statistique d'un texte (coefficient 4, durée ≈ 40 min)

Il vous est demandé de concevoir un programme qui effectue l'analyse statistique d'un texte, c'est-à-dire qui calcule la fréquence d'apparition de chaque lettre de l'alphabet dans le texte considéré (un tel programme peut permettre, par exemple, de décrypter un texte codé). On supposera, pour simplifier, que le texte est composé de lettres minuscules uniquement.

Soit, par exemple, le texte suivant :

```
voici le texte
```

Le programme devra calculer la fréquence d'apparition des lettres et afficher :

```
frequence du caractere c = 8.33 %
frequence du caractere e = 25.00 %
frequence du caractere i = 16.67 %
frequence du caractere l = 8.33 %
frequence du caractere o = 8.33 %
frequence du caractere t = 16.67 %
frequence du caractere v = 8.33 %
frequence du caractere x = 8.33 %
```

Principe de programmation :

- tout d'abord, le programme doit lire au clavier le texte à analyser ;
- puis, il calcule le nombre d'apparitions des lettres de l'alphabet de "a" à "z" et le nombre total N de lettres qui composent le texte ;
- enfin, il affiche la fréquence d'apparition de chaque lettre présente.

Conseils de programmation :

- Déclarez un variable T du type chaîne de caractères pour le texte à analyser et une variable A du type tableau d'entiers pour stocker le nombre d'apparitions des lettres de l'alphabet.
- Sachant que le code ASCII de la lettre "a" est 97 et celui de la lettre "z" est 122, vous pouvez indicer le tableau A de 97 à 122.
- Utilisez les fonctions prédéfinies **length** et **chr** du Pascal.

PROBLEME : jeu de Mastermind (coefficient 6, durée $\approx$ 60 min)

Vous allez programmer le célèbre jeu de Mastermind qui consiste à rechercher une combinaison secrète de quatre couleurs choisies parmi six couleurs. Nous considérons les six couleurs suivantes : rouge (R), bleu (B), vert (V), jaune (J), noir (N) et orange (O).

Principe du jeu :

1. L'ordinateur choisira de façon aléatoire une combinaison secrète de quatre couleurs parmi les six couleurs possibles. Une couleur peut être présente zéro, une ou plusieurs fois. Soit par exemple, la combinaison secrète à rechercher : RJRB.
2. L'utilisateur du programme va faire un premier essai en proposant une combinaison de quatre couleurs, par exemple : VBRJ.
3. Le programme compare l'essai à la combinaison secrète, puis affiche le nombre x de couleurs qui figurent à la bonne place dans l'essai, ainsi que le nombre y de couleurs qui sont présentes mais qui ne figurent pas à la bonne place dans l'essai. Pour l'essai VBRJ, les valeurs de ces deux nombres sont $x = 1$ et $y = 2$.
4. Les étapes 2 et 3 sont répétées jusqu'à ce que l'utilisateur trouve la combinaison secrète, c'est-à-dire lorsque $x = 4$.

Principe de programmation :

- Ecrire une procédure **INITIAL** sans arguments qui génère une combinaison aléatoire C de quatre couleurs en utilisant la fonction **random** du Pascal. C sera la combinaison secrète à trouver. Cette procédure calculera également le nombre de fois où chacune des six couleurs apparaît dans la combinaison C et le stockera dans un tableau T de six entiers. Les variables chaîne de caractères C et tableau T seront déclarées comme variables globales.
- Ecrire une procédure **ESSAI** qui permet à l'utilisateur du programme de proposer une combinaison E. Cette procédure aura pour arguments la chaîne de caractères E ainsi qu'un tableau F de six entiers permettant de comptabiliser le nombre de fois où chacune des six couleurs apparaît dans la combinaison E.
- Ecrire une procédure **RESULTAT** qui compare l'essai E à la combinaison C et affiche les valeurs des nombres x et y définis ci-dessus (voir principe du jeu). Cette procédure aura pour arguments la chaîne de caractères E et le tableau F. Les variables x et y seront déclarées comme variables globales.

TSVP

- Ecrire le programme principal qui fait, tout d'abord, appel à la procédure **INITIAL**, puis ensuite, fait appel aux procédures **ESSAI** et **RESULTAT** jusqu'à ce que l'essai E soit identique à la combinaison recherchée C .

Conseils pour le calcul de x et y :

- Le calcul de x dans la procédure **RESULTAT** ne présente pas de difficulté, il suffit de comparer les chaînes de caractères E et C .
- En revanche, le calcul de y dans la procédure **RESULTAT** est plus complexe. Il vous est conseillé pour cela de comparer le contenu des tableaux F et T .

Exemple d'exécution du programme :

On suppose, par exemple, que le programme a généré la combinaison secrète RJRB. Au cours de la recherche de cette combinaison par l'utilisateur, votre programme devra afficher successivement les lignes suivantes :

```
Entrez votre essai : VBRJ
Valeurs de x et y pour cet essai : x = 1  y = 2

Entrez votre essai : VRBN
Valeurs de x et y pour cet essai : x = 0  y = 2

Entrez votre essai : JORB
Valeurs de x et y pour cet essai : x = 2  y = 1

Entrez votre essai : BJRB
Valeurs de x et y pour cet essai : x = 3  y = 0

Entrez votre essai : RJRB
Valeurs de x et y pour cet essai : x = 4  y = 0

BRAVO !!!
```