

Initiation à la programmation**Examen d'informatique****2ème session - Février 2006**

durée 2h - documents et calculatrices non autorisés

Ce sujet comprend trois exercices indépendants. Ils peuvent être traités dans n'importe quel ordre.

Exercice 1 : Sur les cartes

Q 1 . On souhaite que les tas de cartes soient remplis de la manière suivante :

- tas 1 : un nombre quelconque de trèfles ; ces cartes sont surmontées par 1 pique.
- tas 2 : un nombre quelconque de carreaux ; ces cartes sont surmontées par 1 ou 2 trèfles.
- tas 3 : un nombre quelconque de cœurs ; ces cartes sont surmontées par 1 carreau.
- tas 4 : un nombre quelconque de piques ; ces cartes sont surmontées par 1 cœur.

Donner une suite d'instructions PASCAL permettant d'obtenir cette situation initiale.

Q 2 . À partir de la situation initiale précédente, on souhaite atteindre la situation suivante (qu'on appelle S_1) :

tas 1 : [T] Tas 2 : [K] Tas 3 : [C] Tas 4: [P]

Donner les instructions PASCAL permettant de passer de la situation initiale à la situation S_1

Q 3 . Voici le code PASCAL d'une fonction nommée UnAutreTas écrite par quelqu'un. Cette fonction n'est pas correcte, elle est censée trouver un tas différent de t1 et t2

```
function UnAutreTas(const t1,t2:TasPossibles):TasPossibles;
var res:TasPossibles;
begin
  if t1 <> 1 then begin
    res:=1;
  end else begin
    if t2 < 4 then begin
      res:=4;
    end else begin
      res:=2;
    end { if };
  end { if };
  UnAutreTas:=res;
end (*UnAutreTas*);
```

Q 3.1. Avec cette fonction, quels sont les affichages produits par les instructions suivantes :

```
writeln(UnAutreTas(1,2));
writeln(UnAutreTas(1,3));
writeln(UnAutreTas(1,4));
writeln(UnAutreTas(2,1));
```

Q 3.2. Quelle correction doit-on apporter au code de la fonction afin que le résultat soit correct ?

Q 3.3. Quelle contrainte d'utilisation à la fonction UnAutreTas doit-on ajouter pour qu'elle fasse réellement ce qui est annoncé ?

Q 3.4. La fonction UnAutreTas telle qu'elle est écrite peut-elle retourner la valeur 3 ?

On souhaite réaliser une procédure nommée MettreAuFond paramétrée par un TasDepart et un TasArrivee qui sont des TasPossibles. Cette procédure prendra une carte située *au sommet*

du TasDepart et la placera *au fond* du TasArrivee c'est à dire, en dessous de toutes les cartes qui se trouvent sur le TasArrivee. Pour plus de simplicité, on impose la contrainte d'utilisation suivante : TasDepart et TasArrivee doivent être différents.

Q 4 . Cette procédure doit elle avoir d'autres contraintes d'utilisation ?

Q 5 . Donner l'entête de la procédure MettreAuFond.

Q 6 . En supposant que la procédure MettreAuFond soit déjà déclarée, expliquez comment on peut l'utiliser pour passer de la situation initiale à la situation suivante :

Tas 1 : C[T] Tas 2 : P[K] Tas 3 : (T+TT)[C] Tas 4 : K[P]

Q 7 . Proposer un algorithme pour MettreAuFond. On utilisera la fonction UnAutreTas en supposant qu'elle a été modifiée, et qu'elle ne comporte plus de contraintes d'utilisation.

Exercice 2 : *Le plus petit*

Q 1 . On veut déterminer le plus petit nombre d'une suite de nombres entiers tapés au clavier un par un, suite achevée dès qu'un nombre négatif est lu. Par exemple, si la suite de nombres tapés est

3.5 1 6 2 -1

le plus petit est 1 (-1 ne compte pas). On supposera que la suite de nombres contient au moins un nombre positif ou nul.

Q 1.1. Déclarez les variables nécessaires à la détermination de ce minimum.

Q 1.2. Donnez la suite d'instruction déterminant le plus petit.

Q 2 . On veut maintenant calculer les deux plus petits entiers lus. On suppose qu'au moins deux nombres positifs ou nuls sont lus.

Q 2.1. Déclarez les variables supplémentaires pour effectuer ce calcul.

Q 2.2. Donnez les instructions permettant de déterminer ces deux plus petits entiers.

Exercice 3 : *Sur les chaînes de caractères*

Le but de cet exercice est de réaliser une fonction de mélange des caractères d'une chaîne. Le mélange souhaité est obtenu

1. en séparant la chaîne s à mélanger en deux chaînes s_1 et s_2 obtenues en coupant la chaîne s en deux :

`timoleon` \mapsto s_1 =`timo` et s_2 =`leon`

Dans le cas d'une chaîne de longueur impaire, s_1 a un caractère de moins que s_2 :

`cunegonde` \mapsto s_1 =`cune` et s_2 =`gonde`

2. puis en mêlant les caractères des deux chaînes obtenues

$s_1 s_2$ =`tliemoon`

$s_1 s_2$ =`cguonnede`

Ainsi le mélange de la chaîne `timoleon` est `tliemoon` et celui de la chaîne `cunegonde` est `cguonnede`.

Q 1 . Réalisez la fonction `fusionne` qui mêle les caractères de deux chaînes comme indiqué ci-dessus.

```
// fusionne(s1,s2) = chaîne dont les
// caractères d'indices impairs sont ceux de s1
// et les caractères d'indices pairs ceux de s2
// CU : s2 est aussi longue que s1 ou
// possède un caractère de plus
function fusionne(s1,s2 : STRING) : STRING;
```

Q 2 . En utilisant la fonction `fusionne`, réalisez la fonction nommée `mélange` pour obtenir le mélange souhaité.