Département STPI Jeudi 9 Avril 2009

I3 - Algorithmique

Durée : *1h30*
Documents autorisés : **AUCUN** (calculatrice comprise)

Remarques :
- Veuillez lire attentivement les questions avant de répondre.
- Le barème donné est un barème indicatif qui pourra évoluer lors de la correction.
- Rendez une copie propre.
- N'utilisez pas de crayon à papier.

1 Questions de cours (5 points)

Répondez aux questions suivantes en 5 lignes maximum :

1. Qu'est ce que le paradigme de la programmation structurée ?
2. Dans le cycle en V, quel est le rôle de la conception préliminaire ?
3. Pourquoi dit-on que les types tableaux sont des types complexes ?
4. Quelle(s) condition(s) doit vérifier deux tableaux $t1$ et $t2$ pour que l'expression $t1 = t2$ soit vrai ?
5. Quel est le rôle du caractère « ; »(point virgule) dans un programme Pascal ?

Solution proposée :

1. Sous paradigme du paradigme de la programmation imprative avec le concept de sous-programme. Cela induit les concepts de paramètres formels, paramètres effectifs et passage de paramètre. Il existe deux types de sous-programme les fonctions (calculent des valeurs) et les procédures (modifie l'état du programme).
2. Choisir un paradigme de programmation.
3. Plusieurs valeurs peuvent être stockées par une seule variable.
4. Outre le faite que $t1$ et $t2$ doivent être du même type, $t1 = t2 \Leftrightarrow \forall i, t1[i] = t2[i]$.
5. Séparateur d'insctruction

2 Développement limité (5 points)

Lorsque x est proche de 0, $sin(x)$ peut être approximé à l'aide de la formule suivante :

$$\sum_{i=1}^{n} \frac{(-1)^i}{(2i+1)!} x^{2i+1}$$

Complétez le corps de la fonction suivante qui calcule une approximation de $sin(x)$ jusqu'au rang n. Utilisez uniquement les variables et paramètres donnés et n'utilisez aucune autre fonction (pas d'analyse descendante) :

fonction sin (x : **Reel**, n : **Naturel**) **: Reel**

Déclaration i : **Naturel**
numerateur,denominateur,xPuissance2IPlus1 : **Reel**
resultat : **Reel**

debut
...
fin

Solution proposée :

fonction sin (x : **Reel**, n : **Naturel) : Reel**

Déclaration i : **Naturel**
numerateur,xPuissanceI : **Reel**
resultat : **Reel**

debut
numerateur ← 1
denominateur ← 1
xPuissance2IPlus1 ← x
resultat ← 0
pour i ←1 **à** n **faire**
numerateur ← -1*numerateur
denominateur ← denominateur*(2*i+1)*(2*i)
xPuissance2IPlus1 ← x*x* xPuissance2IPlus1
resultat ← resultat+numerateur/denominateur*xPuissanceI
finpour
retourner resultat
fin

3 Majuscule (10 points)

Dans le cours présentant des programmes de chiffrement, nous avons considéré posséder la fonction $majuscule$ qui permet de calculer à partir d'une chaîne de caractères ch une chaîne de caractères ch' tel que tous les caractères minuscules, et uniquement eux, de ch soient transformés en majuscule dans ch'. La signature de cette est fonction est :

– **fonction** majuscule (uneChaine : **Chaine de caracteres) : Chaine de caracteres**

Ainsi $majuscule(" abc, ?ABC")$ donne la valeur $" ABC, ?ABC"$.

L'objectif de cet exercice est de donner l'algorithme de cette fonction en considérant que nous avons les trois fonctions suivantes :

– **fonction** longueur (uneChaine : **Chaine de caracteres) : Naturel**
– **fonction** iemeCaractere (uneChaine : **Chaine de caracteres**, position : **Naturel) : Caractere**
– **fonction** caractereEnChaine (unCaractere : **Caractere) : Chaine de caracteres**

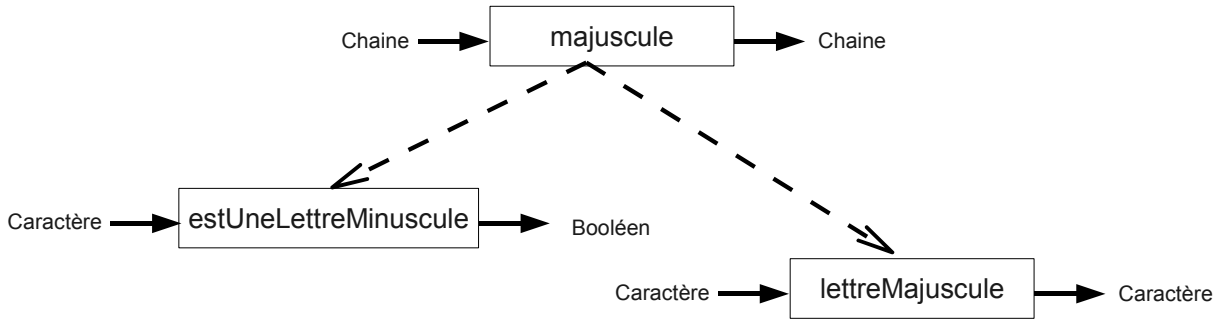
3.1 Analyse

Pour calculer la version majuscule d'une chaîne de caractères ch, on a besoin de savoir calculer la majuscule d'un caractère c de ch lorsque c représente une lettre minuscule. Nous n'avons aucun a priori concernant la table de codage de ces caractères, si ce n'est que :

– le caractère 'a' précède le caractère 'b', qui précède le caractère 'c', etc.
– le caractère 'A' précède le caractère 'B', qui précède le caractère 'C', etc.

Proposez une analyse descendante de ce problème à l'aide du formalisme vu en cours.

Solution proposée :



3.2 Conception préliminaire

Déterminez la signature des fonctions ou procédures correspondant aux opérations de votre analyse descendante.

Solution proposée :

- **fonction** majuscule (ch : **Chaine de caracteres) : Chaine de caracteres**
- **fonction** estUneLettreMinuscule (c : **Caractere) : Booleen**
- **fonction** lettreMajuscule (c : **Caractere) : Caractere**

3.3 Conception détaillée

Donnez le corps de chacune de ces fonctions ou procédures.

Solution proposée :

fonction majuscule (ch : **Chaine de caracteres) : Chaine de caracteres**

Déclaration i : **Naturel**
c : **Caractere**
resultat : **Chaine de caracteres**

debut
resultat ← ””
pour i ←1 **à** longueur(ch) **faire**
c ← iemeCaractere(ch,i)
si estUneLettreMinuscule(c) **alors**
resultat ← resultat+ caractereEnChaine(lettreMajuscule(c))
sinon
resultat ← resultat+ caractereEnChaine(c)
finsi
finpour
retourner resultat
fin
fonction estUneLettreMinuscule (c : **Caractere) : Booleen**
debut
retourner c≥’a’ et c≤’z’
fin
fonction lettreMajuscule (c : **Caractere) : Caractere**
debut
retourner chr(ord(’A’)+ord(c)-ord(’a’))
fin
ou

fonction lettreMajuscule (c : **Caractere) : Caractere**

Déclaration i : **Caractere**
resultat : **Caractere**

debut
resultat ← 'A'
pour i ←'b' **à** c **faire**
resultat ← succ(resultat)
finpour
fin

3.4 Développement

Donnez le code Pascal de la fonction $majuscule$.

Solution proposée :

```
function majuscule(ch : String) : String;
var
   resultat : String;
   i        : Integer;
   c        :  Char;
begin
   resultat:='';
   for i:=1 to length(ch) do
   begin
      c:=ch[i];
      if estUneLettreMinuscule(c) then
         resultat:=resultat+lettreMajuscule(c)
      else
         resultat:=resultat+c
   end;
   majuscule:=resultat
end; { majuscule }
```