Premier cycle Vendredi 23 Mai 2008

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.

1 Questions de cours (5 points)

Répondez aux questions suivantes en 5 lignes maximum :

1. Citez (en les expliquant) 4 critères permettant d'évaluer un bon programme.
2. Qu'est ce que le paradigme de la programmation impérative ?
3. Dans le cycle en V, quel est le rôle de la conception détaillée ?
4. Dans le cycle en V, quel est le rôle des tests unitaires ?
5. Qu'est-ce que l'encapsulation ?

Solution proposée :

1. lisible, fiable, maintenable, réutilisable, portable, correct (preuve), efficace (complexité)
2. Programme = succession d'états (commencer à l'état initial pour finir à état final). état = ensemble des variables. changement d'état = utilisation instruction d'affectation
3. Donner corps aux fonctions et procédures de la conception préliminaire
4. Vérifier que les fonctions et procédures fonctionnent correctement
5. Le fait de ne pas utiliser (ne pas connaître) la façon dont un type est codé et de n'utiliser que les fonctions et procédures proposées pour manipuler les variables de ce type

2 Développement limité (5 points)

Lorsque x est proche de 0, $ln(1+x)$ peut être approximé à l'aide de la formule suivante :

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

Complétez le corps de la fonction suivante qui calcule une approximation de $ln(1+x)$ jusqu'au rang n. Utilisez uniquement les variables et paramètres donnés et n'utilisez aucune autre fonction (pas d'analyse descendante) :

fonction ln1PlusX (x : **Reel**, n : **Naturel) : Reel**

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

debut
...

fin

Solution proposée :

fonction ln1PlusX (x : **Reel**, n : **Naturel) : Reel**

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

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

3 *estUnPrefixe* (6,5 points)

Pour cet exercice on considère que l'on possède les fonctions suivantes :

- **fonction** longueur (uneChaine : **Chaine de caracteres) : Naturel**
  ...avec $longueur("") = 0$
- **fonction** iemeCaractere (uneChaine : **Chaine de caracteres**, iemePlace : **Naturel) : Caractere**
  ...avec $0 < iemePlace \leq longueur(uneChaine)$

Proposez la fonction *estUnPrefixe* qui permet de savoir si une première chaîne de caractères est préfixe d'une deuxième chaîne de caractères (par exemple « pré » est un préfixe de « prédire »).

Solution proposée :

fonction estUnPrefixe (lePrefixePotentiel,uneChaine : **Chaine de caracteres) : Booleen**

Déclaration i : **Naturel**
resultat : **Booleen**

debut
si longueur(lePrefixePotentiel)>longueur(uneChaine) **alors**
retourner FAUX
sinon
i ← 1
resultat ← VRAI
tant que resultat et i≤longueur(lePrefixePotentiel) **faire**
si iemeCaractere(uneChaine,i)=iemeCaractere(lePrefixePotentiel,i) **alors**
i ← i+1
sinon
resultat ← FAUX
finsi
fintantque
retourner resultat
finsi
fin

4 Jeu des 7 erreurs (3,5 points)

Le programme pascal suivant comporte 7 erreurs de syntaxe. Identifiez les en donnant à chaque fois le numéro de la ligne, l'erreur commise et la solution.

```
1  program nbparfait
2
3  function estUnDiviseur(a,b : Integer) : Boolean;
4  begin
5     estUnDiviseur:=a mod b=0
6  end; { estUnDiviseur }
7
8  function sommeDesDiviseurs(n :  Integer) : Integer;
9  var somme,i : Integer;
10 begin
11    somme=0;
12    for i:=1 to n div 2 do
13       if estUnDiviseur(n,i) then
14          somme:=somme+i;
15    sommeDesDiviseurs:=somme
16 end; { sommeDesDiviseurs }
17
18 function estParfait : Boolean;
19 begin
20    estParfait:=(n=sommeDesDiviseurs(n))
21 end; { estParfait }
22
23 fonction obtenirBorneMax();
24 var borneMax :  Integer;
25 begin
26    write("Valeur maximale d''affichage des nombres parfaits : ");
27    readln(borneMax);
28    obtenirBorneMax:=borneMax
29 end; { obtenirBorneMax }
30
31 procedure afficherNombresParfaitsJusquA(n : Integer);
32 var i :  Integer;
33 begin
34    for i:=1 to n do
35       if estParfait(i) then
36          write(i,' ');
37    writeln('')
38 end; { afficherNombresParfaitsJusquA }
39
40 procedure afficherDesNombresParfaits();
41 var nb :  Integer;
42 begin
43    nb:=obtenirBorneMax();
44    afficherNombresParfaitsJusquA(nb)
45 end; { afficherDesNombresParfaits }
46
47 begin
48    afficherDesNombresParfaits()
49 end
```

Solution proposée :

1. ligne 1 : manque ; (*nbparfait ;*)
2. ligne 11 : manque un : (*somme :=0 ;*)
3. ligne 18 : manque le paramètre de la fonction est parfait ((*n : Integer)*)
4. ligne 23 : manque le type de retour (*: Boolean*)
5. ligne 23 : fonction à la place de function
6. ligne 26 : remplacer les ' par "

7. ligne 49 : manque le . final