

Série de révision

Ex1:(23 mai 2007 9h):

Soit l'algorithme du programme principal suivant:

- 0) Début traitement
- 1) Ecrire ("donner le nbre d'éléments du tableau")
Lire (N)
- 2) Pour i de 1 à n faire
 Ecrire ("T [", i, "]=")
 Lire (T[i])
Fin pour
- 3) Proc decale (N, T)
- 4) Proc affiche (N, T)
- 5) Fin traitement

L'algorithme ci-dessus est celui d'un programme qui permet de

- décaler toutes les chaînes non vides au début du tableau T en gardant leur ordre à l'aide de la procédure decale
- afficher les chaînes non vides du tableau T en utilisant la procédure affiche

Exemple: soit N=7 et le tableau T suivant:

T	Souris	Clavier		Ecran		Disquette	Scanner
i	1	2	3	4	5	6	7

Après le programme traitement, le tableau T devient:

T	Souris	Clavier	Ecran	Disquette	Scanner		
i	1	2	3	4	5	6	7

Questions:

1. Traduire l'algorithme "traitement" en pascal
2. ajouter le contrôle nécessaire à la séquence 1 pour que $2 \leq N \leq 15$
3. transformer les séquences 1 et 2 en une procédure "lecture" et prévoir son appel par le programme principal
4. écrire la procédure decale permettant de décaler toutes les chaînes non vides au début du tableau T en gardant leur ordre
5. écrire la procédure affiche permettant d'afficher les chaînes non vides du tableau T après traitement

Ex2:(23 mai 2007 14h):

Ecrire un programme pascal qui permet de saisir une chaîne non vide de longueur impaire et de l'afficher sous la forme d'un sablier

Exemples:

Ch="SABLIER"

```

S A B L I E R
  A B L I E
    B L I
      L
        B L I
          A B L I E
            S A B L I E R

```

Ch="ECRAN"

```

E C R A N
  C R A
    R
      C R A
        E C R A N

```

Ex3:(23 mai 2007 15h30):

Un entier est dit sublime si la somme de ses diviseurs y compris lui-même et le nombre de ses diviseurs sont deux nombres parfaits.

Un entier est dit parfait s'il est égal à la somme de ses diviseurs sauf lui-même

Exemple:N=12

- la somme des diviseurs SD de 12 est $1+2+3+4+6+12=28$
- le nombre de diviseurs NBD de 12 est 6
- on a
 - 28 est parfait car $28=1+2+4+7+14$
 - 6 est parfait car $6=1+2+3$

=>donc 12 est sublime

On propose les algorithmes suivants:

0) Début sublime

1) Ecrire ("N=")

Lire (N)

2) Proc diviseur (N, SD, NBD)

3) Proc affiche (N, SD, NBD)

4) Fin sublime

0) Def fn parfait (m:entier):booléen

1) Somd<-1

Pour i de 2 à (m div 2) faire

Si (m mod i)=0 alors

Somd<-somd+i

Fin si

Fin pour

2) parfait<- (somd=m)

3) Fin parfait

Questions:

0. traduire l'algorithme sublime en pascal

1. écrire la procédure "diviseurs" permettant de calculer la somme S des diviseurs d'un entier p y compris lui-même et le nombre NB de ces diviseurs

2. écrire la procédure "affiche" permettant de vérifier si un nombre p est sublime ou non. Cette procédure utilisera la fonction "parfait" dont l'algorithme est mentionné ci-dessus

Ex4:

Ecrire une analyse et un algorithme intitulé "Annuaire" qui permet de remplir 2 tableaux Tnom et Ttel en parallèle en respectant que à chaque nom de Tnom correspond le numéro du téléphone de même indice

Exemple:

Tnom	Walid	Kais	Mohammed	Ramzi	Abdelaziz
Ttel	92765324	27895312	75650222	90456032	29034598
i	1	2	3	4	5

Questions:

1. écrire un module "remplir_Tnom" qui permet de remplir le tableau Tnom en respectant que un nom est composé seulement de lettres majuscules
2. écrire un module "remplir_Ttel" qui permet de remplir le tableau Ttel en respectant qu'un numéro de téléphone est composé exactement de 8 chiffres et commence obligatoirement par 7(tel fixe) ou 9(GSM Tnitel) ou 2(GSM Tunisiana)
3. écrire un module "Tri_nom" qui permet de trier les deux tableaux en parallèle suivant l'ordre croissant des noms
4. écrire un module "Tri_tel" qui permet de trier les deux tableaux en parallèle suivant l'ordre croissant des numéros
5. écrire le programme principal qui permet d'appeler les modules ci-dessus et saisir un nom pour déterminer son numéro de téléphone s'il existe ou saisir un numéro de téléphone pour connaître le nom de son propriétaire en utilisant la recherche dichotomique

ex5:(bac pratique 2001):

Soit la suite $(P_i)_{i \text{ impair}}$ définie par :

$$\begin{cases} P_1 = 2 \\ P_i = P_{i-2} \times \frac{i-1}{i} \times \frac{i+1}{i} \end{cases} \quad (i > 1 \text{ et } i \text{ impair})$$

Ecrire un programme Pascal qui permet de calculer et d'afficher les termes de la suite P jusqu'à ce que la différence entre deux termes consécutifs devient inférieure ou égale à 10^{-4} .

N.B : La solution doit comporter au moins une fonction et une procédure.

Ex6:(Bac blanc 2006):

Soit la fonction suivante :

Function verifier(ch:string):boolean;

var

 i:integer;

 trouve:boolean;

Begin

 i:=0;

 repeat

 i:=i+1;

 trouve:= ch[i] in ['A'..'Z'];

 until (trouve=false) or (i=length(ch));

 verifier :=trouve ;

End;

1. Convertir cette fonction en une procédure.
2. Le résultat de l'appel est affecté à une variable B, compléter les instructions suivantes permettant l'affectation du résultat à B.

Appel de la fonction verifier Write (B);	Appel de la procédure verifier Write (B);
--	---

3. Soit la procédure ORDRE suivante : Ecrire l'équivalente de cette procédure sous forme de fonction :

```

Procédure ORDRE ( c : Char ; VAR i : Integer) ;
Begin
If Uppcase ( c ) = c Then  i := ORD ( c ) - ORD ( 'A' ) +1
                          Else  i := ORD ( c ) - ORD ( 'a' ) +1 ;
End ;
    
```

4. Le résultat de l'appel est affecté à une variable x, compléter les instructions suivantes permettant l'affectation du résultat à x.

Appel de la procédure	Appel de la fonction
..... Write(x) ; Write(x) ;

5. Trouver le résultat de l'exécution à la main renvoyé par la fonction ORDRE pour les caractères suivants sachant que ORD ('A') = 65 et ORD ('a') = 97 :

'D'
.....

'e'
.....

'F'
.....

'a'
.....