Rattrapage - 2018/2019

Exercice 1 : (4 Pts)

1- Ecrire une fonction récursive qui calcule le PGCD de deux entier strictement positifs A et B.
2- Donner la complexité de cet algorithme.

Exercice 2 : (5 Pts)

Soit CH une chaine de caractère contenant un texte avec des sous chaines entre parenthèses.

Exemple : CH= '27 ja(ivn)er rat(rt)apa()(glA eg)o'

1- Ecrire une fonction **Verifier** permettant de vérifier si l'ordre des parenthèses dans la chaine est correct.
2- Au fait, les sous chaines entre parenthèses sont inversées. Ecrire une procédure **Corriger** qui corrige la chaine CH en supprimant les parenthèses et en remettant les sous chaines concernées en ordre.
   Exemple : CH corrigée : '27 janvier rattrapage Algo'

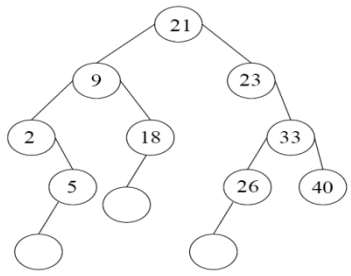
Exercice 3 : (11 Pts)

Soit **Premier** une fonction booléenne à un paramètre entier qui vérifie si un entier donnée est premier.

1- Ecrire une fonction **PremierMin** qui renvoie le premier ***nombre premier*** strictement supérieur à un entier A (A>0).
2- Soit X un entier strictement positif. Nous voulons réaliser une décomposition en produit de facteurs premiers de X (ex : $X=1078000 = 2^4x5^3x7^2x11$). Pour sauvegarder cette décomposition, on propose d'utiliser une liste chainée, ou chaque élément de la liste représente un facteur (le nombre premier et sa puissance).
   a- Donner la déclaration de cette liste (**Tliste**)
   b- Ecrire une procédure **Decompose**, permettant de créer une liste représentant la décomposition d'un entier X.
   c- En utilisant cette décomposition, écrire une fonction **CarreParfait** permettant de vérifier si une telle liste représente un entier carré parfait.
   d- Ecrire une procédure **Racine2** permettant de créer une liste représentant la décomposition de la racine carrée d'un entier carré parfait (**il n'est pas demandé de calculer la racine carrée**).
3- Soit T un tableau de N entiers positifs ($N \leq 100$), et soit **Tfile** un type de files où chaque élément est une liste représentant la décomposition d'un entier.
   a- Donner la déclaration de **Tfile.**
   b- Ecrire un algorithme permettant de créer une file contenant les représentations des éléments carrés parfaits du tableaux T, puis supprimer de la file obtenue toutes les représentations des entiers pairs.

Question Bonus (1Pt)

En choisissant des valeurs adéquates complétez l'arbre binaire de recherche ci-contre.



Bonne Chance

Correction

Exercice 1 :

1-

```
Fonction PGCD(A,B :entier) :entier ;
Debut
        Si  A mod B=0   Alors PGCD ← A
                        Sinon PGCD ← PGCD(B,A mod B);
       Fsi;
Fin ;
```

2- Complexité :

Le nombre d'appels de la fonction va dépendre des valeurs de *A* et *B*. Au pire des cas tous les deux appels le nombre A (ou B) diminue de moitié ($A=2^k$, B=2 ou bien $B=2^k$, A=2), dans ce cas on fait **k** appels.

Donc la complexité de l'algorithme est O(Max(Log(A),Log(B)).

Exercice 2 : 5 Pts

1- Fonction verifier :

```
Fonction Verifier(CH :chaine) :booleen ;
Var     P:Pile;   x :caractère;       T,I :entier;
Debut
        I←Taille(CH) ; Verifier←Vrai ; Initpile(P) ;
        Tantque I≤T et Verifier
        Faire   Si CH[I]='('
                Alors Empiler(P,CH[I])
                Sinon   Si CH[I]=')'
                        Alors   Si Pilevide(P) Alors Verifier←Faux
                                Sinon Depiler(P,x)
                                Fsi
                        Fsi
                 Fsi ;
                I←I+1 ;
        Fait ;
        Si Non Pilevide(P) Alors Verifier←Faux Fsi;
Fin ;
```

2- Procédure Corriger :

```
Procedure Corriger(E/S  CH :chaine) ;
Var CHI :chaine ; x :caractère ;  P:Pile;
Debut
        Si Verifier(CH)
        Alors   T ←Taille(CH) ; Initpile(P) ;  CHI←'' ; I←1 ;
                Tantque I≤T
                Faire   Si CH[I]≠'('      Alors  CHI←CHI+CH[I]
                                          Sinon  I←I+1 ;
                                                 Tantque CH[I]≠')' Faire Empiler(P,CH[I]) ; I←I+1 ; Fait ;
                                                 Tantque Non Pilevide(P)
                                                 Faire Depiler(P,x) ; CHI←CHI+x ; Fait
                        Fsi;
                        I←I+1 ;
                Fait ;
                CH←CHI ;
        Fsi;
Fin ;
```

Exercice 3 : 11 Pts.

1- Fonction PremierMin :

```
Fonction PremierMin(A :entier) :entier ;
Debut
```

```
        A←A+1 ;
        Tantque Non Premier(A)  Faire A←A+1 ; Fait ;
        PremierMin←A ;
Fin ;
```

2-

a- Déclaration

```
Tliste=^Liste ;
Liste= Enregistrement   Nbp,Ps :entier; Suiv :Tliste ; FinEnreg ;
```

b- Procédure Decompose :

```
Procedure Decompose(E/ X :entier ; S/ LP:Tliste) ;
Var     P,Q :Tliste ;  M,A,Ps :entier ;
Debut
        Si Premier(X)
        Alors   Allouer(LP) ; LP^.Nbp←X ; LP^.Ps←1 ; LP^.Suiv←Nil
        Sinon   //créer la tete
                A←1 ; A←PremierMin(A) ;         Ps←0 ;
                //recherche 1ere diviseur
                Tantque X mod A ≠0 Faire  A←PremierMin(A) ;  Fait ;
                Tantque X mod A =0 Faire  X←X div A ; Ps←Ps+1 ; Fait ; //calcul 1ere facteur
                Allouer(LP) ; LP^.Nbp←A ; LP^.Ps←Ps ; P←LP ;
                //créer les autres facteurs
                M←X div 2 ;
                Tantque A≤M
                Faire   A←PremierMin(A) ;  Ps←0 ;
                        Tantque X mod A =0 Faire        X←X div A ; Ps←Ps+1 ; Fait ;
                        Si Ps≠0
                        Alors   Allouer(Q) ; Q^.Nbp←A ; Q^.Ps←Ps ;
                                P^.Suiv←Q ; P←Q ;
                                M←X div 2 ;
                        Fsi ;
                Fait ;
                //traitement dernier X premier
                Si Premier(X) Alors  Allouer(Q) ; Q^.Nbp←X ; Q^.Ps←1 ; P^.Suiv←Q Fsi ;
                P^.Suiv←Nil
        Fsi ;
Fin ;
```

c- Fonction CarreParfait :

```
Fonction CarreParfait(LP :Tliste) :booleen ;
Debut
        CarreParfait←Faux;
        Si LP≠Nil
        Alors   CarreParfait←Vrai;
                Tantque LP≠Nil et CarreParfait
                Faire   Si LP^.Ps mod 2=1 Alors  CarreParfait←Faux Fsi;
                        LP←LP^.Suiv ;
                Fait ;
        Fsi ;
Fin ;
```

d- Procedure Racine2

```
Procedure Racine2(E/ LP :Tliste; S/ LR :Tliste) ;
Var     P,Q :Tliste ;
Debut
        LR←Nil ;
        //créer LP en FIFO
        Si LP≠Nil et CarreParfait(LP)
        Alors   Allouer(LR) ; LP^.Nbp←LP^.Nbp ; LR^.Ps←LP^.Ps  div 2 ; P←LR ;
                LP←LP^.Suiv ;
                Tantque LP≠Nil
```

```
                Faire   Allouer(Q) ;
                        Q^.Nbp←LP^.Nbp ; Q^.Ps←LP^.Ps  div 2 ;
                        P^.Suiv←Q ; P←Q ;
                        LP←LP^.Suiv ;
                Fait ;
                P^.Suiv←Nil
        Fsi ;
Fin ;
```

3-

a- Déclaration

```
TFile= Enregistrement   Tete,Queue :Efile ; FinEnreg ;
Efile=^CelluleF ;
CelluleF= Enregistrement   Info :Tliste ; Suiv :Efile ; FinEnreg ;
```

b-

```
Algorithme FilePremier ;
Type ----
Var     T :Tableau[1..100] de entier ;
        I,N :entier ;
        L :Tliste ;
        F1,F2 :Tfile ;
        //Declaration des différentes actions paramétrées
Debut
        Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
        Pour I←1 à N Faire Lire(T[I]) ; Fait ;
        //Traitement
        InitFile(F1) ;
        Pour I←1 à N
        Faire   Decompose(T[I],L) ;
                Si CarreParfait(L) Alors Enfiler(F1,L) Fsi ;
        Fait ;
        InitFile(F2) ;
        Tantque Non FileVide(F1)
        Faire   Defiler(F1,L) ;
                Si L^.Nbp≠2 Alors Enfiler(F2,L) Fsi ;
        Fait ;
        Tantque Non FileVide(F2)
        Faire   Defiler(F2,L) ;  Enfiler(F1,L) Fait ;
Fin.
```

Question Bonus :

Premier nœud : 2≤ Etiq <5, on peut prendre 2,3,4
Deuxième nœud : 9≤ Etiq <18, on peut prendre 9,10,11,12,13,14,15,16,17
Troisième nœud : 23≤ Etiq <26, on peut prendre 23,24,25