Licence Informatique
Algorithmique Avancée
TD 3: Octobre 2006

1. (a) Combien de multiplications sont utilisées par l'algorithme donné en cours pour calculer x^n quand $n = 32$? Peut-on calculer x^{32} avec moins de multiplications?
   (Au temps pour moi; je voulais poser cette question pour la version de l'algorithme sans multiplications inutiles par 0)
   Réponse 5; non.

   (b) Mêmes questions pour $n = 30$.
   Réponse 7; oui (par exemple calculer les puissances 2, 3, 6, 9, 15, 30).

   (La version en exalgo:)

```
fonction puissance(val x:reel; val i: entier): reel;
var p:reel;
debut
   si i=1 alors retourne x
   sinon
      p:=puissance(x,i/2);
      si i%2=0 alors retourne p*p
      sinon retourne p*p*x
      finsi
   finsi
fin
finfonction;
```

2. Ecrire une fonction récursive `pgcd(m,n)` qui calcule le plus grand diviseur commun des deux entiers (non-négatifs) `m` et `n`.

   Réponse

```
fonction pgcd (val m,n : entier):entier;
debut
   si n=0 alors retourne(m); finsi
   retourne(pgcd(n, m % n));
fin
finfonction
```

3. Une procédure `P(n,T)` a pour paramètre un entier et un tableau unidimensionnel et n'utilise que les éléments `T[1],...,T[n]`. Je veux appeler `P(n,T)` une fois avec chaque combinaison possible de valeurs des éléments `T[1],...,T[n]` avec la contrainte que `T[i]` prend une valeur entre 1 et `B[i]` ($1 \leq i \leq n$) et la somme de tous les `T[i]` est au maximum K. Il est à noter que n est une variable.

(a) Ecrire une procédure récursive en `exalgo` qui effectue tous ces appels.
Réponse

```
procedure Appliquer(val i,sommemax:entier; ref T: tab);
debut
   si i=0 alors P(n,T);
   sinon
      T[i]=1;
      tantque T[i]<=B[i] et T[i]+(i-1)<=sommemax faire
         Appliquer(i-1, sommemax-T[i], T);
         T[i]:=T[i]+1
      fintantque
   finsi
fin
finprocedure
```

et `Appliquer(n,K,T);` fait ce qui est souhaité.

(b) Et une procédure non-récursive.

```
procedure Appliquer(val n:entier; ref T: tab);
var i:entier;
var somme:entier;
debut
   somme:=n;
   pour i allant de 1 a n par pas de 1 faire T[i]:=1; finpour;
   repeter
      P(n,T);
      i:=1;
      tantque i<=n et (T[i]=B[i] ou somme=K) faire
         somme:=somme-T[i]+1;T[i]:=1;i:=i+1
      fintantque;
      si i<=n alors T[i]:=T[i]+1;somme:=somme+1 finsi
   jusqu'a i>n finrepeter
fin
finprocedure
```