

DS2 - Algorithmes et Structures de Données

Mardi 14 Janvier 2020

Durée 3H – Documents non autorisés

1. Primitive d'un polynôme - 5 pts

On souhaite écrire une fonction primitive $F(x)$ d'un polynôme.

Exemple :

$$p(x) = x^3 + 2x^2 + 4x + 2$$

$$F(x) = \frac{1}{4} x^4 + \frac{2}{3} x^3 + 2x^2 + 2x$$

Ecrire en pseudo-langage la fonction suivante : Fonction primitive ($p : \text{poly}$) : poly
pour chacune des représentations suivantes :

1.1. Le polynôme est représenté par un tableau de réels où les indices (commençant à 0) représentent les degrés des termes et les réels les coefficients. **Faire un dessin pour l'exemple**

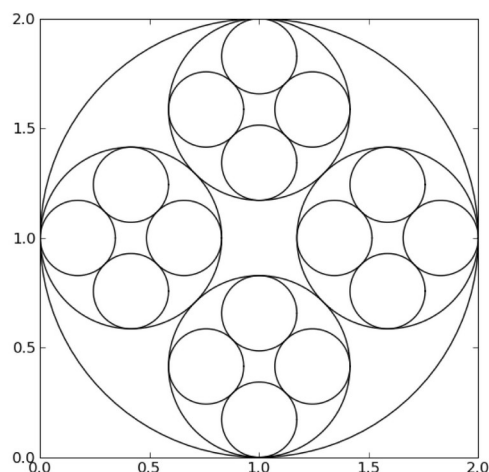
Type $\text{poly} = \text{tableau } [0..20] \text{ de réels}$

1.2. Le polynôme est représenté par une liste chaînée. **Faire un dessin pour l'exemple**

Type $\text{poly} = \text{^terme}$
 $\text{terme} = \text{Enregistrement}$
 $d : \text{entier}$
 $c : \text{réel}$
 $\text{suiv} : \text{^terme}$
 FinEnregistrement

2. Dessin récursif - 5 pts

On souhaite dessiner à l'écran la figure dont on a mis ci-dessous la 3^{ème} étape.



Soit la procédure `TraceCercle` ($\underline{E} \ x, y, r : \text{entier}$) qui dessine un cercle à l'écran de centre de coordonnées (x, y) et de rayon r . Ecrire en pseudo-langage la **procédure récursive** `cercles` qui trace tous les cercles dont le rayon est suffisamment grand.

Procédure `cercles` ($\underline{E} \ x, y, r : \text{entier}$)

A chaque étape, les rayons sont divisés par $1 + \sqrt{2}$

Le premier cercle est de centre $(100, 100)$ et de rayon $r=100$ (en pixels).

3. Table de hachage - 10 pts

On souhaite créer une table de hachage des mots d'un texte contenu dans un fichier. Cette table est un tableau (dont les indices (clés) sont numérotés de 1 à c) de c listes chaînées. La fonction de hachage $h(m)$ associe au mot m un entier correspondant à la clé de ce mot (par exemple, le numéro de la première lettre du mot ($a=1$)). Les mots de la $i^{\text{ème}}$ liste sont les mots m de l'ensemble donnant $h(m)=i$ comme clé de hachage. **Les c listes sont triées dans l'ordre alphabétique des mots.**

Les structures de données utilisées sont les suivantes :

```

Const  c  = 300
Type   liste = ^cellule
          cellule = Enregistrement
                  nom : chaîne
                  suiv : liste
                  FinEnregistrement
          tab = tableau[1..c] de liste
Var    t : tab

```

De plus, on a la fonction suivante :

```

Fonction h (m : chaîne) : entier

```

3.1. Ecrire en pseudo-langage une procédure insère qui insère un mot m dans la table t (l'insertion n'est pas effectuée si m appartient déjà à t). **Faire des dessins.**

```

Procédure insère (E m : chaîne ; E/S t : tab, E c : entier)

```

3.2. Ecrire en pseudo-langage une procédure crée qui remplit la table à partir des mots d'un fichier texte. On pourra utiliser la procédure mot-suiv qui donne en sortie le mot m d'une chaîne de caractères c à partir de la position i (mot-suiv(c, i, m) a été écrite en TD, i est incrémenté de façon à ce qu'on puisse lire le mot suivant).

```

Procédure crée (E nomfich : chaîne ; S t : tab, E c : entier)

```

3.3. Ecrire en pseudo-langage une procédure sauvegarde qui enregistre la table dans un fichier. Chaque ligne du fichier correspond à une liste de la table (on considère que les listes contiennent suffisamment peu de mots pour tenir sur une seule ligne), selon la syntaxe suivante :

```

clé1 :: mot11, mot12, ...
clé2 :: mot21, mot22, ...

```

```

Procédure sauvegarde (E t : tab, E c : entier)

```