ULCO **2019 - 2020**
Licence 2 **Mathématiques**

Examen – Informatique – Session 2

Documents autorisés, pas de livre, pas de calculatrice
Lundi 13 janvier 2020
Durée : 2h

Exercice 1. [5 points] Nous nous proposons de représenter les vecteurs de $\mathbb{R}^n$ à l'aide de la structure suivante

```
struct Vecteur{
   array<Reel,n> coeffs;
};
```

1. **[1 pt]** Écrire une fonction `bool est_nul(Vecteur u)` qui teste si le vecteur `u` est le vecteur nul.

Soit a un réel positif. On définit la suite $(u_n(a))_{n\in\mathbb{N}}$ par récurrence en posant $u_0(a) = 1$ et

$$u_{n+1}(a) = \frac{1}{2}\left(u_n(a) + \frac{a}{u_n(a)}\right).$$

On montre que la suite $(u_n(a))$ converge vers $\sqrt{a}$.

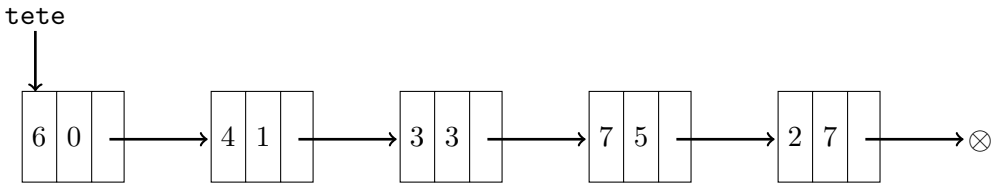
2. **[1.5 pts]** Écrire une fonction `reel racine(reel a)` qui retourne u_{100}(`a`), valeur approchée de $\sqrt{a}$.
3. **[1 pt]** Écrire une fonction `reel norme(Vecteur u)` qui retourne une approximation de la norme de `u`. Vous utilizerez la fonction `racine` contruite à la question précédente.
4. **[1.5 pts]** Écrire une fonction `Vecteur somme(Vecteur u,Vecteur v)` qui prend en paramètre deux vecteurs `u` et `v` et retourne le vecteur `u + v`.

Exercice 2. [7 points] Dans cet exercice nous représentons des polynômes de $\mathbb{R}[X]$ à l'aide de liste chaînée dont les éléments correspondent à des monômes :

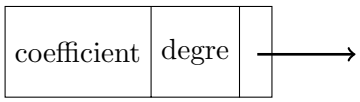
```
struct Polynome{
    Monome* tete;
};
```

```
struct Monome{
    Monome* suivant;
    Reel coefficient;
    int degre;
};
```

Nous souhaitons que les coefficients soient rangés par degré croissant. Par exemple le polynôme $2\,X^7 + 7\,X^5 + 3\,X^3 + 4X + 6$ sera représenté par la liste chainée



où une cellule contient :



On supposera pour la suite que tous les monômes constituant une telle liste chaînée possède un coefficient non nul. En particulier le polynôme nul sera représentée par la liste vide.

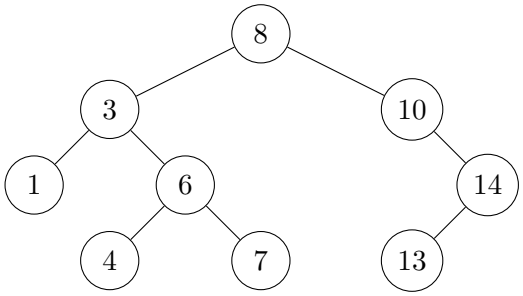
1. **[1 pt]** Écrire une fonction `int degre(Polynome P)` retournant le degré du polynôme `P`. Le degré du polynôme nul étant $-\infty$ par convention, la fonction retournera -1 dans ce cas précis.
2. **[2 pts]** Écrire une fontion `Reel evalue(Polynome P,Reel a)` qui retourne le réel `P(a)`.
3. **[2 pts]** Écrire une fonction `Polynome derive(Polynome P)` qui retourne le polynôme $\mathtt{P}'$ dérivé du polynôme `P`.
4. **[2 pts]** Écrire une fonction `Reel dominant(Polynome P)` qui retourne le coefficient du monôme de degré maximal du polynôme `P`.

Exercice 3. **[8 points]** Dans cet exercice les arbres binaires seront représentés par la structure suivante :

```
struct Arbre{
   Arbre* gauche;
   Arbre* droite;
   Reel valeur;
};
```

Un *arbre binaire de recherche* est un arbre binaire A tel qu'en chaque noeud n de A on ait :
- la valeur de la racine du sous-arbre gauche de n (s'il existe) est strictement inférieure à la valeur du noeud n ;
- la valeur de la racine du sous-arbre droit de n (s'il existe) est strictement supérieure à la valeur du noeud n.

L'arbre suivant est un tel arbre binaire de recherche.



1. **[1 pt]** Donner l'ordre de visite des noeuds de l'arbre précédent lors des parcours préfixé, infixé et postfixé. Que remarquez vous de particulier ?
2. **[1.5 pts]** Écrire une fonction `Reel minimum(Arbre* A)` qui étant donné un arbre binaire de recherche non vide `A` retourne le plus petit réel présent dans `A`. Que faut-il changer pour obtenir le plus grand réel présent dans `A` ?
3. **[1.5 pts]** Écrire une fonction `bool rechecher(Arbre* A,Reel v)` qui étant donné un arbre binaire de recherche `A` retourne `true` si le réel `v` est présent dans `A` et `false` sinon.
4. **[2 pts]** Écrire une fonction `bool est_abr(Arbre* A)` qui retourne `true` si l'arbre binaire `A` est un arbre binaire de recherche et `false` sinon.
5. **[2 pts]** Écrire une fonction `void ajoute(Arbre* A,Reel v)` qui ajoute la valeur `v` à l'arbre de recherche `A` si `v` n'est pas déjà présent dans `A` ; vous ne ferez pas appel à la fonction `rechercher`. Nous rappellons que le code `new Arbre` retourne un pointeur sur un élément de type `Arbre`. **Attention** l'arbre ainsi obtenu devra toujours être un arbre binaire de recherche.