Université de Batna 2
Département d'informatique
Licence 2 : informatique académique

Algorithmique et Structures de Données
29 janvier 2017

Contrôle Final

Durée 1h30

Exo1 (4 pts) : la complexité

- La fonction ci-dessous permet de calculer X^n « **puissance (x,n)** ».

```
double puissance (double x, int n){

    double resultat=1 ;

    if (n==0){
            resultat=1 ;
    }else{
            int i ;

            for (i=0 ;i<n ;i++){
                    resultat = resultat*x ;
            }
    }

return resultat ;

}
```

Q1) Calculer la **complexité** de cette fonction.

- L'algorithme ci-dessous permet calculer la valeur d'un polynôme P(x) de degré N pour une valeur X donnée.

$P(x) = 3X^2 - 7X + 1$ → **N=2**
→ **Les coefficients de ce polynôme sont: P [0]=1, P[1] = -7, P[2]= 3**
→ Pour X= 5 → evaluer (2, 5, P) retourne 41.

```
double evaluer (int N, double X, double P []){
        int resultat = 0;
        int i;
        for (i=0 ;i <=N ;i++){

                resultat= resultat + P[i]* puissance (X,i) ;
        }

        return resultat ;
}
```

Q2) Quelle est la complexité de la fonction : **evaluer** ?

Exo2 (12 pts) : liste doublement chainée

Un étudiant est caractérisé par : son numéro d'inscription, son nom (20 caractères), son prénom (20 caractères), et sa moyenne générale.

Q1) Proposer une structure de donnée chainée « **étudiants** » qui permet de gérer les étudiants d'une manière **bidirectionnelle**.

Q2) Si un pointeur occupe un espace mémoire de 4 octets. Donner le résultat de la fonction : **sizeof (étudiant) ;**

Q3) Présenter avec un schéma de pointeurs **les 3 cas de suppression** d'un élément d'une **liste doublement chainée** qui contient 4 éléments

Q4) Ecrire une fonction qui permet **d'insérer** un étudiant dans une **liste doublement chaînée triée**.

Remarque 1 : La liste **doit restée triée** par ordre croissant du *N*.
Remarque 2 : on pourra pas y avoir 2 étudiants qui ont le même numéro.

Exo3 (4 pts): Piles + Arbre binaire de recherche

On veut trier un tableau d'entier et le placer dans une pile **en utilisant un arbre binaire de recherche**.

Q1) Décrire les étapes à suivre afin de réaliser cet objectif.

Q2) Quel type de parcours devons nous utiliser ? Et pourquoi ?

Q3) Ecrire **les fonctions** nécessaires qui permettent de trier les éléments d'un tableau et les placer dans une pile (toujours en utilisant un arbre binaire de recherche).

Remarque : dans l'exercice 3, vous pouvez utiliser directement les fonctions vues dans le cours :

- **InsererABR (A, X) :** insérer l'élément X dans l'arbre binaire de recherche A

- **Pile_vide (P1)** =1 si la pile vide, 0 si la pile contient au moins un élément.
- **X= Dépiler (P1).**
- **Empiler (P1, X).**

- **File_vide (F1)** =1 si la file vide, 0 si la File contient au moins un élément.
- **X= Défiler (F1).**
- **Enfiler (F1, X).**

- **Etc...**

Bon courage seulement aux étudiants qui travaillent dur