Université de Carthage **Année Universitaire 2013/2014**

Ecole Supérieure de la Statistique et de l'Analyse de l'Information

Module : Algorithmique et programmation II

Enseignantes : Mme Aïcha El Golli Jabbes/Sihem Mansour

Niveau d'études : 1ère année **Durée de l'épreuve :** 1h30

A. Récursivité, structures et pointeurs (barème envisagé : environ 4 points, 10-15 minutes)

Soit la procédure itérative suivante :

```
void AFF (int n )
{
        int i ;
        for( i=1 ;i<= n ;i++)
                printf("#") ;
}
```

1. Exécutez cette procédure avec n=3 ### **(1point)**
2. Ecrire la version récursive en langage C de cette procédure.

```
void AFF (int n ) 0.5point
{   0.25point
  if(n>0)                  0.5point
        {  0.25point

                printf("#") ; 0.5point
                AFF(n-1);  0.5point
        }  0.25point

} 0.25point
```

B. Structures et pointeurs (barème envisagé : environ 3 points, 5-10 minutes)

On considère les types et les déclarations suivantes :

```
typedef struct c{
        int valeur [40];
        struct c *Suivant ;
        }complexe ;
```

```
typedef struct{
        complexe* debut ;
        int n ;
        }classes ;
```

```
complexe T  [20];
 classes C ;
```

Quel est le type de ces expressions :

(a) T[i] **complexe (0.5pt)**
(b) T[i].valeur[j] **entier (0.5pt)**
(c) T[i].suivant **pointeur vers complexe(0.5pt)**
(d) C.debut **pointeur vers complexe (0.5pt)**
(e) C.debut→valeur[i]**entier (0.5pt)**
(f) C.debut→suivant→valeur[i] **entier (0.5pt)**

C. Liste chaînée (barème envisagé : environ 9 points, 30-40 minutes)

1. Soit la définition suivante d'une liste linéaire chaînée :

```
Typedef struct Cel{
int valeur  ;
Struct cel*Suivant  ;
}cellule ;
```

Typedef struct Liste* cellule ;

a. Ecrire une fonction en langage C **Max (l : Liste) : entier** qui retourne le maximum d'une liste.

(2points)

Entête (**0.25point)**

Si liste vide (**0.5point)**

Sinon initialisation max (**0.25point)**

Boucle sur les elts ; comparaison et mise à jour (**0.5point)**

Retour du max (**0.5point)**

```
int Max(Liste l)
{ int m;
   if(l==NULL)return -1;
      else
   {
      m=l->valeur;
      l=l->suiv;
      while(l!=NULL)
      {
         if(m<l->valeur) m=l->valeur;
         l=l->suiv;
      }
      return m;
   }
}
```

b. Ecrire une procédure en langage C **SuppDernier (d/r l : Liste)** qui supprime le dernier élément d'une liste. **(2.5points)**

```
void SuppDernier(Liste* l){
   cellule* p=(*l);
   cellule* q;
   if(*l!=NULL)
   {
      if(p->suiv!=NULL) // cas de +eurs elt dans la liste
         {
      while(p->suiv!=NULL){
      q=p;
      p= p->suiv;}
      q->suiv=NULL;
      free(p);
      }
    else // Cas où il y a un seul elt dans la liste
         {
                  *l=NULL;
                  free(p);
         }
   }
}
```

c. Ecrire une procédure en langage C **Fusion(**d/r **l1 : Liste,** d/r **l2 : Liste)** qui fusionne la liste l2 avec la liste l1 et l2 devient vide.

(2.5points)

```
void Fusion(cellule** l1, cellule** l2)
{
   if(*l1!=NULL)
   {
```

```
        if(*l2!=NULL)
        {
           cellule* p=*l1;
           while(p->suiv!=NULL)
           {
              p=p->suiv;
           }
           p->suiv=*l2;
           *l2=NULL;
        }
    }
    else
        *l1=*l2;
}
```

d. Ecrire une fonction en langage C récursive **Longueur** qui calcule le nombre d'éléments de la liste. **(2points)**

```
int Longueur(Liste l)
{
    if (l==NULL) return 0;
    else
        return (1+ Longueur(l->suiv));
}
```

D. Piles et files (barème envisagé : environ 4 points, 15-25 minutes)

1. Quelle est la différence entre une pile et une file ? **(1points)**
2. Soit la définition suivante d'une file :

```
typedef struct element
{
    int donnee;
    struct element *suivant;
} Elem;
typedef struct file
{
Elem *tete;
Elem *queue;
}File;
```

a. Écrire la fonction int sommet(File F) en langage C, retournant la valeur du premier élément, mais sans le défiler **(2points)**

```
int sommet(File F)
{
    int ret = -1;
    if (F.tete != NULL)
    {
        ret = F.tete->donnee;
    }
   return ret;
}
```

b. En vous basant sur les primitives vues en cours, dessiner l'état de la mémoire après la suite des instructions suivantes :

creefile(F); enfiler(F, 5); enfiler(F,3); enfiler(F,2); defiler(F); enfiler(F,sommet(F)); enfiler(F,sommet(F)); defiler(F);

2 3 3 (1points)

Bon Courage