Examen d'Algorithmique - (2010/2011 - session 2)

Durée : 1 heure

Formation Ingénieurs CNAM

Ce sujet comporte 2 pages et les questions sont largement indépendantes. On s'attachera à soigner la présentation du code afin qu'il soit le plus lisible possible. Il ne suffit en aucun cas d'écrire le code d'une fonction, il faut expliquer (i.e. commenter) les choix faits pour l'implantation. Il est de plus indispensable de respecter les notations données dans l'énoncé.

Exercices sur les listes

On rappelle la structure de données définie pour représenter les listes d'entiers ainsi que les opérations de base `vide` et `cons`. Afin de pouvoir tester notre implantation, on programme également une fonction d'affichage `print`.

```
#include<stdio.h>

typedef struct liste
{
  int x;
  struct liste *suivant;
} Sliste, *Liste;

Sliste *vide()
{ return (NULL); }

Sliste *cons(int v, Sliste *l)
{
  Sliste *m=(Sliste *)malloc(sizeof(Sliste));
  m->x=v;
  m->suivant=l;
  return m;
}

void print(Sliste *l)
{
  while (l!=NULL)
    {
      printf("%d ",l->x);
      l=l->suivant;
    }
  printf("\n");
}
```

Question 1 Programmer la fonction `plus1` (qui ajoute 1 à chaque élement de la liste). On procèdera de manière itérative (en utilisant une boucle `while`) avec des effets de bord (le type de retour de la fonction demandée est `void`).

```
void plus1 (Sliste *l)
{ ... }
```

Question 2 Programmer de manière récursive une fonction `somme` qui fait la somme de tous les éléments d'une liste.

```
int somme(Sliste *l)
{ ... }
```

Question 3 Programmer une fonction qui inverse l'ordre des éléments dans une liste. On pourra utiliser une fonction intermédiaire `ajout_fin` qui ajoute un élément à la fin d'une liste.

```
Sliste* ajout_fin(Sliste *l, int u)
{  ... }

Sliste* reverse (Sliste *l)
{ ... }
```

Question 4 Ecrire un programme de test dans la fonction `main` qui produit la sortie suivante :

```
int main()
{ ... }
```

```
user@computer$ ~/test-listes
3 6 8
somme = 17
4 7 9
somme (+1)= 20
reverse (4 7 9) = (9 7 4)
user@computer$
```

Question 5 Plutôt que de faire la somme des éléments d'une liste, on choisit de faire les soustractions successives. Par exemple, pour la liste (4 7 9), on souhaite effectuer les opérations suivantes : (4 - 7) - 9. Dans cet exemple, le résultat attendu est donc -12.

```
int sous(Sliste *l, int acc)
{ ... }
```