Année universitaire 2016-2017 – L2 informatique
Code UE : 21IM32
Durée : 2 heures

Le sujet comporte trois pages.

Consignes : aucun document autorisé, calculatrices et téléphones portables interdits, dictionnaires papiers autorisés.

Architecture des systèmes
Examen terminal

Partie 1 – Circuits logique (7 points)

Le but de cette partie est la réalisation d'un circuit logique permettant de tester si deux entiers *A* et *B* représentés en RBNS (Représentation binaire non signée) sur *2* bits, sont différents.

Exercice 1.1 : Donner la fonction booléenne sous les deux formes (somme des produits et produit des sommes) associée à ce circuit.

Exercice 1.2 : Simplifier la fonction obtenue précédemment (sous la forme de votre choix).

Exercice 1.3 : Donner le schéma du circuit logique correspondant, noté DIFF2.

Exercice 1.4 : Proposer un schéma pour le circuit logique (utilisant les circuits DIFF2 comme des boîtes noires) permettant de tester si deux entiers *A* et *B* représentés en RBNS sur *2k* bits, sont différents.

Partie 2 – Programmation assembleur (7 points)

Exercice 2.1 (4 points) : Programmez en assembleur la fonction suivante, sans changer l'algorithme :

```
void f( ) {
   int i = 0;
   int j = 100;
   while ((i<20) || (j>=50)) {
      j = j-2*i;
      if (j<10) {
         j = j+120;
      }
      i = i+1;
   }
   return;
}
```

Exercice 2.2 (3 points) : Le programme assembleur suivant devrait afficher le message “Total : 34 euros” et quitter. Cependant, ce programme comporte six erreurs. Pour chaque erreur : identifiez la, expliquez le fonctionnement de ce programme si l'erreur n'est pas corrigée, et corrigez la.

```
section .data

message db 'Total : 00 euros', 10
length equ $-message

section .text

global _start
_start:
        mov esi, [message]
        add esi, 7
        mov [esi], '3'
        inc esi
        mov al, [esi]
        add ax, '4'
        mov [esi], al
        mov eax, 4
        mov ebx, 1
        mov ecx, message
        mov edx, length
        int 80
        mov eax, 1
        xor ebx, ebx
        int 80h
```

Partie 3 – Programmation avancée en C (6 points)

La cryptographie moderne est basée sur la possibilité de manipuler des nombres premiers très grands, c'est-à-dire, de l'ordre de plusieurs centaines de bits (256 bits ou même 1024 bits). Malheureusement, la taille maximale des entiers non signés en C est bornée par 64 bits ou 128 bits. Nous allons dans cet exercice proposer une bibliothèque pour manipuler des entiers arbitrairement longs.

Exercice 3.1 (1 point) : Pour représenter les entiers nous avons la possibilité soit de les représenter en base 2 soit de les représenter en base 10. Comme nous ne faisons que des opérations en base 2, nous allons préférer une représentation en base 2. Proposez un type `entier_long` qui nous permettra de manipuler des entiers (positifs ou négatifs) de taille arbitraire en base 2.

Exercice 3.2 (1 point) : Ecrivez une fonction qui prend en entrée une chaîne de caractères composée de chiffres en base 10 et qui retourne l'objet correspondant de type `entier_long`. Lorsque le premier caractère est `'-'`, alors l'entier correspondant est négatif, sinon il est positif, c'est-à-dire, il est positif si le premier caractère est `'+'` ou s'il est absent. Vous prendrez en compte les caractères qui ne sont pas des chiffres.

Exercice 3.3 (1 point) : Expliquez comment on pourrait écrire une fonction qui prend en entrée un `entier-long` et qui retourne l'entier correspondant en base 10 sous forme de chaîne de caractères.

Exercice 3.4 (1 point) : Ecrivez une fonction qui prend en entrée un `entier-long` représentant un entier x et qui retourne un `entier-long` représentant $2x$.

Exercice 3.5 (1 point) : Il faut maintenant inclure des fonctions d'addition et de soustraction. Proposez une fonction d'addition qui prend en entrées deux paramètres x et y de type `entier-long` et qui retourne un `entier-long` représentant $x+y$. En déduire une fonction qui fait la soustraction de deux `entier-long`.

Exercice 3.6 (0.5 point) : Expliquez une méthode pour générer de façon aléatoire des `entier-long` premiers.

Exercice 3.7 (0.5 point) : Il faut maintenant faire le choix du type de bibliothèque. Quelle est la différence entre une bibliothèque statique et une bibliothèque dynamique ? Vous choisiriez quel mode pour notre module de manipulation des entiers arbitraires (justifiez) ?