14 avril 1999

UNIVERSITÉ DU QUÉBEC À HULL
DÉPARTEMENT D'INFORMATIQUE

TECHNIQUES DE PROGRAMMATION I
INF4033

EXAMEN FINAL

Directives:

1. C'est un examen à livre ouvert.
2. Répondre directement sur ce formulaire.
3. Utiliser le verso de page comme brouillon.
4. L'examen dure trois heures.
5. Le barème est montré au début de chaque question.
6. Questions éventuelles doivent être posées durant les 30 premières minutes de l'examen.

Nom, prénom:____

Code permanent:____

Signature:____

Numéro du formulaire:____________________________________

Ex.#	1	2	3	4	5	6	7	8	9	10	11	Σ
Note												
sur	8	5	10	11	8	9	8	8	13	8	12	100

Ex. #1 (8 pts)

Pour chacune des expressions en C données ci-dessous, produire son arbre de décomposition:

a) `a = a += b *= a++ > b ? a + b : a - b ;`

b) `-a + b - a + a * b / a + - b;`

Ex. #2 (5 pts)

Donner les valeurs de chaque expression de l'ex. #1 en supposant les déclarations:

```
int a = 8;
double b = 13.0
```

a)

b)

Ex. #3 (10 pts)

Donner la sortie du programme suivant:

```
#include <stdio.h>

int F1 (int x);
int F2 (int y);

main()
{
    int x = 4, y = 7;

    printf ("x au debut de main = %d\n", x);
    y = F2 (x+y);
    printf ("y apres appel de F2 = %d\n", y);
    y = F1 (y+4);
    printf ("y apres appel de F1 = %d\n", y);
}

int F1 (int x)
{
    int y = 4;

    printf ("x au debut de F1 = %d\n", x);
    y = F2(x-1)+3;
    printf ("y au retour de F1 = %d\n", y);
    return (y+2);
}

int F2 (int y)
{
    int x = 3;

    printf ("x au debut de F2 = %d\n", x);
    y = 2*x-3;
    printf ("y au retour de F2 = %d\n", y);
    return (y+3);
}
```

Ex. #4 (11 pts)

Donner la sortie du programme suivant:

```
#include <stdio.h>

int F1 (int x);
int F2 (int y);

int x = 5;

main()
{
    int x = 4, y = 7;

    printf ("x au debut de main = %d\n", x);
    y = F2 (x*y +3);
    printf ("y apres appel de F2 = %d\n", y);
    y = F1 (y+x);
    printf ("y apres appel de F1 = %d\n", y);
}

int F1 (int x)
{

    printf (" au debut de F1 x = %d\n", x);
    x += x + 3;
    return (x = x + 2);
    printf ("x a la fin de F1 = %d\n", x);

}

int F2 (int y)
{

    printf ("au debut de F2 x = %d, y = %d\n", x, y);
    x += y;
    y = 2 *x + y;
    printf ("au retour de F2 x = %d, y = %d\n", x, y);
    return (y+3);
}
```

Ex. #5 (8 pts)

Soient 010101_2, 1011101_2 représentant les nombres binaires, 207_8, 54_8 - les nombres octaux (en base huit) et 37_{10}, 290_{10} - les nombres décimaux.

a) Convertir la valeur de 010101_2 en nombre décimal

Réponse: $\boxed{}$

b) Convertir la valeur de 1011101_2 en nombre octal

Réponse: $\boxed{}$

c) Convertir la valeur de 207_8 en nombre décimal

Réponse: $\boxed{}$

d) Convertir la valeur de 54_8 en nombre binaire

Réponse: $\boxed{}$

e) Convertir la valeur de 37_{10} en nombre binaire

Réponse: $\boxed{}$

f) Convertir la valeur de 290_{10} en nombre octal

Réponse: $\boxed{}$

g) Convertir la valeur de 290_{10} en nombre en base 7

Réponse: $\boxed{}$

h) Convertir la valeur de 207_8 en nombre en base 7

Réponse: $\boxed{}$

Ex. #6 (9 pts)

Donner la sortie du programme suivant:

```
for (i=0; i<14; i+=3){
    j=3;
    while (j<i){
        if ((i+j) % 3 == 2)
            printf ("i = %d; j = %d;\n", i, j);
        j++;
    }
}
```

Ex. #7 (8 pts)

Soit la déclaration

```
    double Tab [17] [30];
```

Supposons que sizeof double == 8, et que le tableau a été alloué à partir de l'adresse de base = 2000.
Dans les cases prévues à cet effet, donner les adresses des éléments suivants:

a) Tab [6]

Réponse: $\boxed{}$

b) Tab [0] [12]

Réponse: $\boxed{}$

c) Tab [13] [0]

Réponse: $\boxed{}$

d) Tab [13] [17]

Réponse: $\boxed{}$

Ex. #8 (8 pts)

Pour le tableau de l'ex. précédent

a) Donner la valeur de sizeof Tab

Réponse: $\boxed{}$

b) Donner la valeur de sizeof Tab [11].

Réponse: $\boxed{}$

c) Quel élément du tableau débute à l'adresse 3200?

Réponse: $\boxed{}$

d) Quel élément du tableau occupe l'octet à l'adresse 4210?

Réponse: $\boxed{}$

Ex. #9 (13 pts)

Dans un programme en langage C, variable text de type string contient une phrase écrite en français (c-à-d une suite de mots séparés par les espaces). Donner un fragment de programme qui calcule le nombre des mots dans ce texte, tels que chacun de ces mots commence et se termine par la même lettre. Vous pouvez vous servir de la librairie strlib.h, vue durant le cours.

Ex. #10 (8 pts)

Soit un tableau
int Tab [100].
Donner un fragment de programme qui calcule la somme de tous les éléments positifs de ce tableau.

Ex. #11 (12 pts)

Donner un fragment du programme qui calcule combien de lignes du tableau
int Tab [N] [M];
contiennent les valeurs négatives. Par exemple, si le tableau contient les valeurs

2	-3	-7	4
4	0	5	10
-2	-5	-3	-8
1	2	3	4
3	-6	9	0

la reponse de votre programme doit être 3 (chaque ligne parmi Tab[0], Tab[2] et Tab[4] contenant au moins une valeur négative).