CSC 3002 - Enoncé du Contrôle Continu de Langage C
Vendredi 9 janvier 2009

Durée : 1 heure
Document autorisé : le polycopié distribué en cours

Exercice 1 (sur 7 points)

Le programme ci-dessous contient plusieurs erreurs et oublis qui font qu'il ne « passe » pas à la compilation. De plus, si on le corrigeait pour qu'il soit compilable, il resterait encore des problèmes à l'exécution.

motDePasse.c

```
#include <stdio.h>
#include <stdlib.h>

int main ( ) {

  char *chaine, motDePasse = "monMotDePasse";

  printf ( "Rentrez votre mot de passe : " );
  scanf ( "%s", chaine );

  if ( chaine == motDePasse )

    printf ( "Vous pouvez passer.\n" );
    return EXIT_SUCCESS;

  else

    printf ( "Désolé, revenez une autre fois.\n" );
    return EXIT_FAILURE;
}
```

Vous devez, **sur la feuille de réponses fournie**, récrire ce programme de façon à ce qu'il soit compilable et qu'il s'exécute correctement.

Exercice 2 (sur 6 points)

Le programme ci-dessous compile et s'exécute normalement. Vous devez simplement écrire, **sur la feuille de réponses fournie**, ce qui s'affiche dans la fenêtre *shell* lorsqu'on l'exécute.

queFaitElle.c (début)

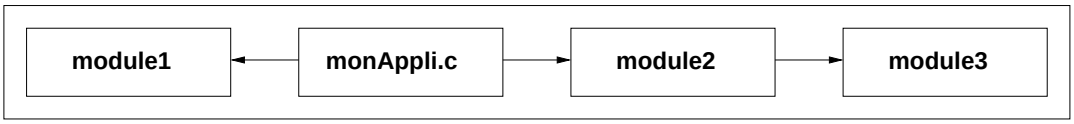
```
#include <stdio.h>
#include <stdlib.h>


int queFaitElle ( int n, int *m );
```

queFaitElle.c (fin)

```
int main ( ) {

  int a = 1, b = 1, c = 1;

  c = queFaitElle ( a, &b );
  printf ( "Après le premier appel de la fonction queFaitElle   : " );
  printf ( "a = %d, b = %d, c = %d\n", a, b, c );
  a = queFaitElle ( b, &c );
  printf ( "Après le deuxième appel de la fonction queFaitElle  : " );
  printf ( "a = %d, b = %d, c = %d\n", a, b, c );
  b = queFaitElle ( c, &a );
  printf ( "Après le troisième appel de la fonction queFaitElle : " );
  printf ( "a = %d, b = %d, c = %d\n", a, b, c );
  return EXIT_SUCCESS;
}

int queFaitElle ( int n, int *m ) {

  *m = *m - 1;
  n = *m * 2;
  return *m + 1;
}
```

Exercice 3 (sur 7 points)

Vous allez devoir créer le fichier *makefile* correspondant à l'application dont l'architecture correspond au schéma ci-dessous.



Comme vous pouvez le voir, cette application est composée d'un programme principal (*monAppli.c*) contenant la définition de la fonction *main* et de trois modules. On rappelle qu'un module est constitué d'un fichier d'en-tête (*.h*) et d'un source C (*.c*). Ainsi le *module1* correspond aux deux fichiers *module1.h* et *module1.c*. Ces sept fichiers se trouvent dans votre répertoire courant.

Une fléche entre deux blocs (programme principal ou module) indique qu'au moins une fonction du bloc extrémité initiale de la fléche appelle une fonction du bloc extrémité terminale de la fléche.

D'autre part le programme principal et les trois modules utilise le type abstrait *Rationnel*. Ce type est implémenté dans la bibliothèque *librationnel.so* dont l'interface est le fichier *rationnel.h*. Le fichier *librationnel.so* est stocké dans le répertoire *lib* sous votre répertoire de connexion. Le fichier *rationnel.h* est stocké dans le répertoire *include* sous votre répertoire de connexion.

Ecrivez, sur la feuille de réponses fournie, le fichier *makefile* pour produire l'application *monAppli*.