

SYSTÈMES D'EXPLOITATION - Devoir Surveillé n°2  
FEUILLE DE REPONSE POUR LA PARTIE 2

Nom, Prénom :

Groupe :

**N'écrire ni au crayon à papier, ni au bic rouge.**

Question 1 : on admet que le script pkg-config fonctionne déjà. Comment peut-on faire pour lancer la commande de compilation comprenant en même temps l'interrogation de la commande pkg-config math, pour que ce qu'elle affiche aille en tant qu'option de cc et qu'on n'ait pas à recopier son affichage lorsqu'on lance cc ? En fait, on veut ne taper qu'une seule ligne qui lance pkg-config math et cc ensemble :

```
prompt% cc monprog.c -o monprog
```

Question 2 : On doit fournir au moins un paramètre à pkg-config : le nom du paquet, sinon c'est une erreur fatale au script. Voir la question 6. Écrire ci-dessous le bout de code qui vérifie cela :

```
if
```

Question 3 : Le nom qu'on a fourni à pkg-config doit être celui d'un paquet, c'est à dire correspondre à un fichier de /usr/lib/pkgconfig qui a l'extension .pc, sinon il faut quitter le script. Vérifier cela pour le premier paramètre passé au script :

```
if
```

Question 4 : On arrive au cœur du script. Admettons que le premier paramètre contienne le nom d'un paquet correct, ex : math. Il faut afficher ce qui est sur la ligne options: de son fichier .pc :

```
if
```

Question 6 : On voudrait maintenant traiter plusieurs paramètres, par exemple si on lance `pkg-config math gtk2.0 opengl`. Que faut-il écrire ? Mettre  $Q_n$  pour dire que le bout de code de la question  $n$  est mis là. On ne reprend pas la question Q5, seulement Q3 et Q4.

Question 7 : Une variante maintenant. Les fichiers `.pc` contiennent une ligne `includes` qui donne une énumération de noms de `.h`, ex : `stdio.h stdlib.h`, ou si vous n'arrivez pas, plusieurs lignes `includes: fichier.h`. Lorsque l'utilisateur tape la commande `pkg-config --includes base`, ça doit lui afficher `#include <stdio.h> #include <stdlib.h> ...` (1 par ligne) au lieu des options de compilation. Comment faire cela ? Ne donner que le cœur du script qui fait cela, pas les vérifications des fichiers et le test de ce paramètre.

Question 8 : Pour aller plus loin, en fait certains paquets ont des dépendances. Par exemple `math` dépend du paquet `base`. Il peut y avoir une liste de plusieurs paquets à la suite, ou si vous n'arrivez pas, plusieurs lignes `depends: autrepaquet`. Donc, finalement quand on tape `pkg-config math`, ça affiche `-Wall -lm` (les options de ses dépendances puis celles de `math`). L'astuce est de rappeler le script lui-même pour chacune des dépendances trouvées pour un paquet : dans `pkg-config`, si le paquet qu'on regarde a une dépendance, alors on relance d'abord `pkg-config` dessus pour avoir ses options et on rajoute ça aux affichages pour le paquet. Écrire le bout de code qui gère cet aspect :