

```
1  jeremy@matterhorn:~/INF123$ tree
2  .
3  |-- a.c
4  |-- a.h
5  |-- b.c
6  |-- b.h
7  |-- c.c
8  |-- d.c
9  |-- d.h
10 |-- e.c
11 `-- e.h
12
13 0 directories, 9 files
14 jeremy@matterhorn:~/INF123$ grep main *.c
15 c.c:int main() {
16 jeremy@matterhorn:~/INF123$ grep include *.c
17 a.c:#include "a.h"
18 a.c:#include "d.h"
19 b.c:#include "b.h"
20 c.c:#include "a.h"
21 c.c:#include "d.h"
22 d.c:#include "d.h"
23 d.c:#include "b.h"
24 e.c:#include "e.h"
```

1. Quel fichier source (.c) contient la routine principale de notre futur exécutable ?
2. Dessiner pour chaque fichier source (.c) la relation d'inclusion avec les fichiers d'en-tête (.h) concernés.
3. Dessiner pour chaque fichier source (.c) les relations de dépendances (et uniquement ces relations) aux fichiers sources et en-têtes (.c et .h) pour générer le fichier objet (.o) correspondant. Utilisez des flèches pour désigner ces dépendances.

Indication : Un fichier source **ne dépend pas** de fichier en-tête. C'est le fichier objet qui dépend du fichier source. Mais comme le fichier source **inclut** (directive de compilation, phase de preprocessing) des fichiers en-têtes, la compilation du fichier objet dépend aussi du contenu des fichiers en-tête, en plus du contenu du fichier source.

4. Pour chaque fichier objet, écrire la commande gcc permettant sa compilation.
5. Écrire pour chaque fichier objet, en utilisant la syntaxe Makefile :
 1. la cible, le fichier objet
 2. ses dépendances
 3. la commande permettant sa génération
6. Ajouter à ce Makefile, en première règle (chaque trio cible, dépendances, commande est une règle), c'est à dire tout en haut du fichier :
 - la règle de compilation finale, combinant tous nos fichiers objets en notre programme exécutable