

1. Que fait le programme suivant ?
2. Décrivez le code et son fonctionnement en détail.
3. Ce programme comporte un défaut majeur, il ne se termine jamais. Pourquoi ?
4. Les lignes 25 et 34 ne sont pas compatibles POSIX. Pourquoi ? Comment y remédier ?

find.c

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <sys/types.h>
4  #include <sys/stat.h>
5  #include <unistd.h>
6  #include <string.h>
7  #include <errno.h>
8  #include <dirent.h>
9  #include <limits.h>
10
11 static void find(const char *dir_name, char *result, int *value ){
12     DIR *d = opendir(dir_name);
13     struct dirent *entry;
14     const char *d_name;
15     if (! d) {
16         fprintf(stderr, "Cannot open '%s': %s\n",
17                 dir_name, strerror(errno));
18         return;
19     }
20
21     while( (entry = readdir(d)) != NULL ) {
22         d_name = entry->d_name;
23         char path[PATH_MAX];
24         snprintf (path, PATH_MAX, "%s/%s", dir_name, d_name);
25         if( entry->d_type & DT_REG ) {
26             struct stat s;
27             if( stat( path, &s ) >= 0 ) {
28                 int current = s.st_size;
29                 if( current > *value ) {
30                     *value = current;
31                     strncpy( result, path, PATH_MAX );
32                 }
33             }
34         } else if( entry->d_type & DT_DIR ) {
35             char res[PATH_MAX];
36             int s = 0;
37             find( path, (char*) res, &s );
38             if( s > *value ) {
39                 *value = s;
40                 strncpy( result, res, PATH_MAX );
41             }
42         }
43     }
44 }
```

```
45     closedir(d);
46 }
47
48
49 int main ( int argc, char **argv ) {
50     int value = 0;
51     char res[PATH_MAX];
52     find( argv[1], (char*) res, &value);
53     printf( "Found: %s (%d).\n", res, value );
54     return EXIT_SUCCESS;
55 }
```