

1. Que fais le programme suivant ?
2. Si le deuxième paramètre du programme (argv[2]) correspond à un lien symbolique, que ce passe-t-il ?
3. Si le dossier passé en premier paramètre (argv[1]) contient des dossiers, que ce passe-t-il ?
4. Si le dossier passé en premier paramètre (argv[1]) contient des liens symboliques, que ce passe-t-il ? Expliquer avec un schéma.

links.c

```
1  #include <sys/types.h>
2  #include <sys/stat.h>
3  #include <dirent.h>
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <unistd.h>
7  #include <errno.h>
8  #include <string.h>
9
10 int main(int argc, char *argv[]) {
11     if (argc != 3) {
12         printf("Invalid number of arguments:\nUsage:\n\t%s source dest\n", argv[0]);
13         return 0;
14     }
15
16     char *in_dir = argv[1];
17     char *out_dir = argv[2];
18
19     struct stat stats;
20     if (stat(out_dir, &stats) < 0) {
21         perror(out_dir);
22         exit(EXIT_FAILURE);
23     }
24
25     if ( !(stats.st_mode & S_IFDIR) ) {
26         fprintf(stderr, "%s: is not a directory\n", out_dir);
27         exit(EXIT_FAILURE);
28     }
29
30     DIR *d = opendir(in_dir);
31     if (! d) {
32         fprintf(stderr, "Cannot open directory '%s': %s\n", in_dir, strerror(errno));
33         exit(EXIT_FAILURE);
34     }
35
36     struct dirent *entry;
37     while( (entry = readdir(d)) != NULL ) {
38         if (strcmp(entry->d_name, "..") != 0 && strcmp(entry->d_name, ".") != 0) {
39
```

```

40     char path_in[PATH_MAX];
41     int res = snprintf(path_in, PATH_MAX,
42                        "%s/%s", in_dir, entry->d_name);
43     if(res >= PATH_MAX)
44         fprintf(stderr, "Input path length is too long.\n");
45
46     char path_out[PATH_MAX];
47     res = snprintf(path_out, PATH_MAX,
48                    "%s/%s", out_dir, entry->d_name);
49     if(res >= PATH_MAX)
50         fprintf(stderr, "Output path length is too long.\n");
51
52     if(link(path_in, path_out) < 0)
53         perror("not possible to link");
54 }
55 }
56
57 if( closedir(d) ) {
58     fprintf(stderr, "Could not close '%s': %s\n", in_dir, strerror (
59         errno));
60     exit (EXIT_FAILURE);
61 }
62 }

```

makefile

```

1 links: links.c
2     gcc links.c -Wall -g -o links

```