TD3: PDS

Exercice 1 : helper:

```
int fillDirs(char **dirs, char *path) {

    int dirs_size = count(':', path) + 1; // \0
    char **dirs = malloc(dir_size * sizeof(char *));
    int i = 0;
    for (int idx=0; path[idx]; ) {
        dirs[i++] = path[idx];
        while (path[idx] != ':' || !path[idx]) idx++;
        path[idx++] = '\0';
    }
    dirs[i] = '\0';
    Return dirs_size;
}
```

```
int count(char c, char* s) {
    int i; int res
    for (i=0; s[i]; i++) {
        if (s[i]==c) res++;
    }
    return res+1;
```

```
int fillDirs(char **dirs){
    char * path;
    if (!(path = getenv("PATH"))) {
        return -1;
    }
    Return fillDirs_helper(dirs, path);
}
```

```
int len(char*s){
    int i=0;
    while(s[i++])
    Return i;
```

Q2.
```
int which(char** dirs, int size, char * command){
    int found=0; int i; char* file;
    for (    i=0; i<size && !found; i++){
        snprintf(file, len(command)+len(dirs[i]), "%s%s",
                 command, dirs[i]);
        if (access(file, F_OK | X_OK)==0){
            found=1;
        }
    }
    if (found) {
        printf("%s", file);
    } else {
        fprintf(stderr, "ERROR: command %s not found", command);
    }
    Return found;
}
```

Q3

```
int main (int argc, char **argv) {
    int success = 0; // success !;   int i
    char * *dirs; int size;
    size = fillDirs (dirs);
    if (argc < 2) {
        fprintf(stderr, "usage : which <path1> .. ");
        exit();
    }

    for (i=1 ; i < argc - 1 ; i++) {
        if (!which (dirs, size, argv[i])) {
            success = 1;
        }
    }
    return success;
}
```