
IR209 - Développement

Rapport de Laboratoire 2: Fonctions itératives vs
fonctions récursives

Maljean Thimothé, Tasiaux Victor

05-10-2022

Table des Matières

1 Exercice 1	3
1.1 Version 1 : Avec variable d'accumulation	3
2 Exercice 2	4
2.1 Arborescence	4
2.2 Algorithme	5
2.3 Code	5

1 Exercice 1

1.1 Version 1 : Avec variable d'accumulation

```
1 #include <stdio.h>
2 void printCharArray(char arr[], int size, int i){
3
4     if (i < size){
5         printf("%c ",arr[i]);
6         printCharArray(arr,4,i+1);
7     }
8 }
9
10
11 int main(void){
12     char arr[] = {'A','B','C','D'};
13     printCharArray(arr,4,0);
14
15     return 0;
16 }
```

Pour se débarrasser du for, la fonction doit s'appeler elle-même. A cette fin, le **if** est ce qui permet de vérifier quand arrêter d'appeler la fonction récursive. ## Version 2: Sans variable d'accumulation

```
1 #include <stdio.h>
2
3
4
5 void printCharArray(char arr[], int size){
6
7
8     if (-size < 0){
9
10         printCharArray(arr,size-1);
11         printf("%c ",arr[size-1]);
12     }
13 }
14
15
16
17 int main(void){
18     char arr[] = {'A','B','C','D'};
19     printCharArray(arr,4);
20
21     return 0;
22 }
```

Pour se débarrasser de la variable d'accumulation, on peut utiliser la variable `size`, qu'on décrémente, ainsi l'utilise comme "compteur" et on a plus besoin de variable `i`. Si on mettait `printCharArray` en fonction suffixe, on listerait le directory en sens inverse. Il faut mettre la fonction `printCharArray` avant le `printf`, donc en fonction préfixe.

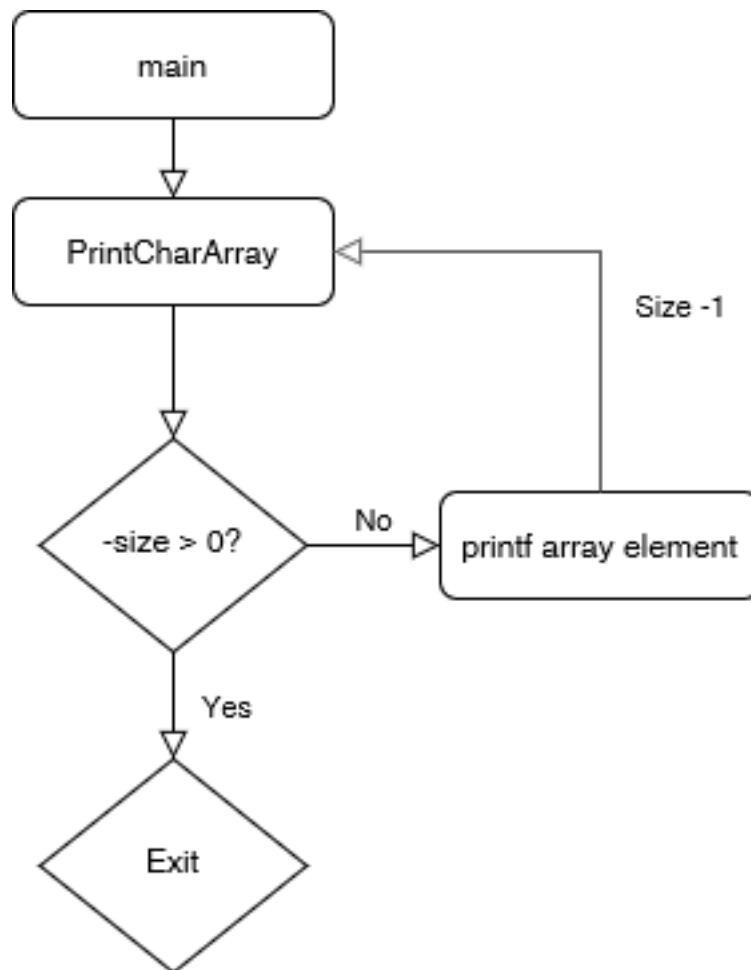


Figure 1: Diagramme de la version 2 de l'exercice 1

2 Exercice 2

2.1 Arborescence

On a créé cette arborescence de test.

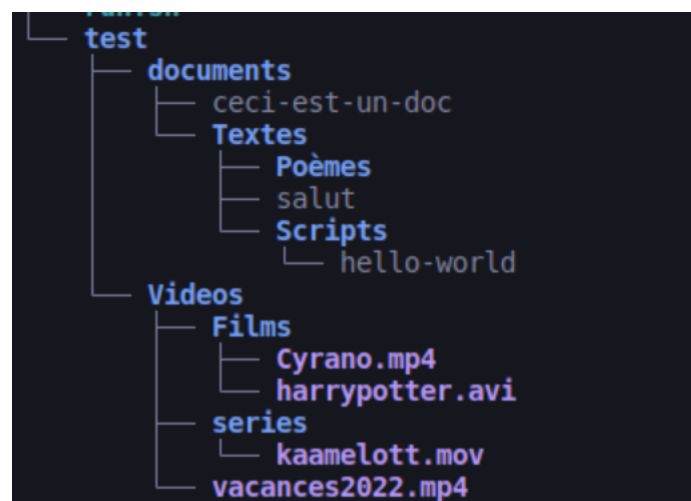


Figure 2: Arborescence de test

2.2 Algorithme

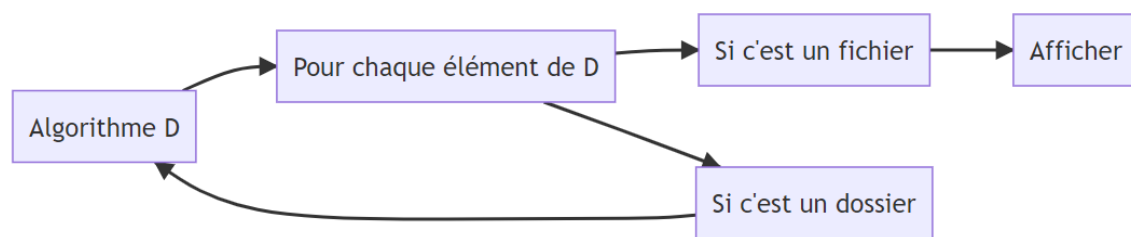


Figure 3: Algorithme utilisé pour créer notre scan d'arborescence

2.3 Code

```

1  #include <dirent.h>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <string.h>
5
6
7  void dirContent(const char rootpath);
8
9
10 int main(void){
11
12     dirContent("/home/kali/Documents/Developpement/Lab2/exercice2/test"
13 );
14
15     return 0;
16 }
17 void dirContent(const char rootpath){
18     puts(rootpath);
19     DIR d = opendir(rootpath);
20     struct dirent* sd = readdir(d);
21
22     while (sd != NULL){
23
24         if(sd->d_type == DT_DIR){
25             char rp, newPath;
26             rp = malloc(sizeof(char) * BUFSIZ);
27             newPath = malloc(sizeof(char) * BUFSIZ);
28             strcpy(rp, d);
29             strcpy(newPath, sd->d_name);
30             strcat(newPath, d);
31             dirContent(newPath);
32         }
33
34         else{
35             if(sd->d_type == DT_REG){
36                 printf("Nom du fichier: %s", sd->d_name);
37             }
38         }
39     }
40 }
41

```

```
42  
43  
44  
45 }
```

A l'heure actuelle, ce code n'est pas fonctionnel.