This question has been translated from Standard ML to OCaml

This question concerns the following OCaml declaration of a tree type:

```ocaml
type 'a fan = Wave of 'a * ('a fan) list
```

(a) Declare the function `flip`, which maps a tree to a mirror image of itself, as illustrated: [3 marks]

```
1
  2  3
   5 6
```

```
1
  4  3
  2 4
```

(b) Declare the curried function `paint f`, which copies a tree while applying the function `f` to each of its labels. [3 marks]

(c) Declare the function `same_shape`, which compares two trees and returns `true` if they are equal except for the values of their labels and otherwise returns `false`. [5 marks]

(d) State the types of functions `flip`, `paint` and `same_shape`. [3 marks]

(e) The function `paper` is declared in terms of the familiar functional `fold_right`:

```
let rec fold_right f l e =
  match l with
  | [] -> e
  | x::xs -> f x (fold_right f xs e)

let rec paper (Wave(x, fs)) q = fold_right paper fs (q + 1)
```

Describe the computation that results when `paper` is applied to a tree. [6 marks]