```
1  //week10
2  // 2015 Dec P 2.2.2
3  (***
4  let rec g1 p xl c =
5      match xl with
6      | x :: xs ->
7          if p x then
8              g1 p xs (fun v -> c (x :: v))
9          else
10             g1 p xs (fun v -> c v)
11     | _ -> c []
12 ***)
13
14 // correct solution
15 let rec g1 p xl c =
16     match xl with
17     | x :: xs when p x -> g1 p xs (fun v -> c (x :: v))
18     | _ -> c []
19
20 g1 (fun x -> x > 3) [ 4; 2; 8; 1; 2; 0; 5 ] id
21
22 let rec g0 p =
23     function
24     | x :: xs when p x -> x :: g0 p xs
25     | _ -> []
26
27 g0 (fun x -> x > 3) [ 4; 2; 8; 1; 2; 0; 5 ]
28
29 // 2011 P3
30 type 'a tree = | Lf
31                | Br of 'a * 'a tree * 'a tree;;
32
33 let rec f(n,t) =
34     match t with
35     | Lf -> Lf
36     | Br(a, t1, t2) -> if n>0 then Br(a, f(n-1, t1), f(n-1, t2))
37                        else Lf;;
38 let t = Br (1, Br (4, Br (3, Br (2, Lf, Lf), Br (5, Lf, Lf)), Br (3, Lf, Lf)),
     Br (7, Lf, Br (4, Br (6, Lf, Lf), Br (9, Lf, Lf))))
39 f(3,t)
40
41 let rec g p =
42     function
43     | Br (a, t1, t2) when p a -> Br (a, g p t1, g p t2)
44     | _ -> Lf;;
45
46 g (fun x -> x < 4) t
47
48 let rec h k =
```

```
49      function
50      | Lf -> Lf
51      | Br(a, t1, t2) -> Br(k a, h k t1, h k t2);;
52  h (fun x -> x*2) t
53
```