```
 1: ::::::::::::::
 2: factorial.ml
 3: ::::::::::::::
 4: (* $Id: factorial.ml,v 361.2 2011-04-25 14:16:06-07 - - $ *)
 5: (* Factorial example. *)
 6:
 7: open Printf;;
 8:
 9: (*
10: let rec fac = function
11:     | 0 -> 1
12:     | n when n > 0 -> n * fac (n - 1)
13:     | n -> invalid_arg ("fac (" ^ (string_of_int n) ^ ")");;
14: *)
15:
16: let fac n =
17:     let rec fac' n' r' = match n' with
18:         | 0 -> r'
19:         | n -> fac' (n' - 1) (n' * r')
20:     in  if n < 0 then invalid_arg ("fac (" ^ (string_of_int n) ^ ")")
21:                  else fac' n 1;;
22:
23:
24: let printfac n = (printf "fac %d = %d\n" n (fac n);
25:                   flush stdout);;
26:
27: printfac 0;;
28: printfac 1;;
29: printfac 2;;
30: printfac 5;;
31: printfac 10;;
32: printfac 20;;
33: printfac (-5);;
34:
35: ::::::::::::::
36: factorial.ml.i
37: ::::::::::::::
38: val fac : int -> int
39: val printfac : int -> unit
40: ::::::::::::::
41: factorial.ml.out
42: ::::::::::::::
43: fac 0 = 1
44: fac 1 = 1
45: fac 2 = 2
46: fac 5 = 120
47: fac 10 = 3628800
48: fac 20 = 2432902008176640000
49: Exception: Invalid_argument "fac (-5)".
```