(\* This is a

multiline, (\* nested \*) comment \*)

type point = { x: float; y: float };;

let some\_string = "this is a string";;

let rec length lst =

match lst with

[] -> 0

| head :: tail -> 1 + length tail

;;

exception Test;;

type expression =

Const of float

| Var of string

| Sum of expression \* expression (\* e1 + e2 \*)

| Diff of expression \* expression (\* e1 - e2 \*)

| Prod of expression \* expression (\* e1 \* e2 \*)

| Quot of expression \* expression (\* e1 / e2 \*)

class point =

object

val mutable x = 0

method get\_x = x

method private move d = x <- x + d

end;;