CSE216
Foundations of Computer Science

Instructor: Zhoulai Fu

State University of New York, Korea

Homework 09

```
(* Exercise 3. Points = 20.

Write a function `seconds_to_time`, of type int -> time, which takes
the seconds elapsed since midnight as its argument and returns the
corresponding time.
*)

let seconds_to_time sec =
    failwith "Not implemented"
```

test code for #3

```
#use "09h.ml";;

let () = assert (seconds_to_time 36610 = {hour = 10;
minute = 10; second = 10})

let () = assert (seconds_to_time 86399 = {hour = 23;
minute = 59; second = 59})

let () = assert (seconds_to_time 86400 = {hour = 0;
minute = 0; second = 0})

let () = assert (seconds_to_time 86410 = {hour = 0;
minute = 0; second = 10})
```

```
(* Exercise 5. Points = 20.

Write a function `tick`, of type time -> time, which increments `t`
by one second and returns the new time:
*)

let tick t =
    failwith "Not implemented."
```

test code for #5

```
#use "09h.ml";;

let () =
let t1 = {hour = 10; minute = 10; second = 10} in
let t2 = {hour = 10; minute = 10; second = 11} in
assert (tick t1  = t2)

let () =
let t1 = {hour = 10; minute = 10; second = 59} in
let t2 = {hour = 10; minute = 11; second = 0} in
assert (tick t1  = t2)

let () =
let t1 = {hour = 10; minute = 59; second = 59} in
let t2 = {hour = 11; minute = 0; second = 0} in
assert (tick t1  = t2)

let () =
let t1 = {hour = 23; minute = 59; second = 59} in
let t2 = {hour = 0; minute = 0; second = 0} in
assert (tick t1  = t2)
```

Homework 10

```
(* Exercise 3 (Points = 20)
Write the function print : exp -> string , which returns a string
representing `e`. The string should print arithmetic operators using
infix notation and properly parenthesize expressions. Your solution
may be similar to the following examples.

print (Add (Int 10, Int 5)) produces "(10 + 5)"
print (Mul (Add (Int 2, Int 3), Int 5)) produces "((2 + 3) * 5)"
print (Mul ((Mul (Int 3, Int 0)), Mul (Int 3, Int 5))) produces "((3
* 0) * (3 * 5))"
*)

let rec print (e:exp):string = failwith "Not Implemented"
```

test code

```
#use "10h.ml";;

let () =
 assert ( print (Add (Int 10, Int 5)) = "(10 + 5)"
 ||  print (Add (Int 10, Int 5)) = "(10+5)"
 )

let ()=
  assert ( print (Mul (Add (Int 2, Int 3), Int 5)) = "
  ((2 + 3) * 5)" ||
  print (Mul (Add (Int 2, Int 3), Int 5)) = "((2+3)*5)"
  )

let () =
  assert ( print (Mul ((Mul (Int 3, Int 0)), Mul (Int
  3, Int 5))) = "((3 * 0) * (3 * 5))" ||
  print (Mul ((Mul (Int 3, Int 0)), Mul (Int 3, Int
  5))) = "((3*0)*(3*5))"
    )
```

Homework 11

Exercise 1. (points = 25)

Write a function drop : int -> 'a list -> 'a list such that drop n lst returns all but the first n elements of lst.
If lst has fewer than n elements, return the empty list. Here, n can be any integer including negative number.

Mock midterm 2

- Real midterm 2 will cover the same topic

- But content can be different in real midterm2