ソフトウェア技法　第１回　201311367　佐藤良祐

課題 1.1

```
(* 課題 1.1 2次方程式の実数解の個数を返す *)
let numRoots (a, b, c) =
  let d = ((b *. b) -. (4.0 *. a *. c)) in
  if d > 0.0 then 2
  else if d = 0.0 then 1
  else 0;;
```

```
# numRoots (2.0, 4.0, -4.0);;
- : int = 2
# numRoots (1.0, 2.0, 1.0);;
- : int = 1
# numRoots (1.0, 0.0, 1.0);;
- : int = 0
```

課題 1.2

```
(* 課題 1.2 時間の変換 *)
let minite2time time =
  let oneday = 60 * 24 in
  let day = time / oneday in
  let hour = (time - (time / oneday) * oneday) / 60 in
  let min = time - (day * oneday + hour * 60) in
  (day, hour, min)

let tuple2time (day, hour, min) =
  day * 24 * 60 + hour * 60 + min

let timeSum ((day1, hour1, time1), (day2, hour2, time2)) =
  minite2time (tuple2time(day1, hour1, time1) + tuple2time(day2, hour2, time2))
```

```
# minite2time 8000;;
- : int * int * int = (5, 13, 20)
# tuple2time (5, 13, 20);;
- : int = 8000
# timeSum ((1,10,30), (4,20,20));;
- : int * int * int = (6, 6, 50)
```

課題 1.3

```
(* 課題 1.3 フィボナッチ数列 *)
let rec fib n =
  if n = 0 then 0
  else if n = 1 then 1
  else fib(n-2) + fib(n-1)
```

```
# fib 10;;
- : int = 55
# fib 15;;;;
- : int = 610
```

```
# fib 12;;
- : int = 144
# fib 0;;
- : int = 0
# fib 1;;
- : int = 1
# fib 2;;
- : int = 1
# fib 3;;
- : int = 2
```

課題 1.4

```
(* 課題 1.4 累乗の計算 *)
(* 方針 1 *)
let rec power1 (x, k) =
  if k = 0 then 1
  else if k = 1 then x
  else x * power1(x, k-1);;

(* 方針 2 *)
let rec power2 (x, k) =
  if k = 0 then 1
  else if k mod 2 = 0 then power2(x * x, k / 2)
  else x * power2(x * x, k / 2);;
```

```
# power1 (2, 8);;
- : int = 256
# power2 (2, 8);;
- : int = 256
# power1 (2, 11);;
- : int = 2048
# power2 (2, 11);;
- : int = 2048
# power1 (2, 30);;
- : int = 1073741824
# power2 (2, 30);;
- : int = 1073741824
```