

プログラミング基礎 実習資料 第 05-06 時限

Haskell を用いて、以下の問題に解答せよ。

問題 1 (A 問題). 2 つの集合の和集合を求める関数 `union` を定義せよ。また、関数 `union` を用いて下記の和集合を求め、その結果を示せ。

- 空集合と空集合の和集合
- 空集合と集合 $\{3, 5, 1, 7\}$ の和集合
- 集合 $\{3, 5, 1, 7\}$ と空集合の和集合
- 集合 $\{2, 4, 5\}$ と集合 $\{3, 5, 1, 7\}$ の和集合

解答例. 以下の通り.

```
union([], bs) = bs
union(a:as, bs) = if memberOf(bs, a) then union(as, bs)
                  else a:union(as, bs)

union([], []) = []
union([], [3,5,1,7]) = [3,5,1,7]
union([3,5,1,7], []) = [3,5,1,7]
union([2,4,5], [3,5,1,7]) = [2,4,3,5,1,7]
```

□

問題 2 (A 問題). `union([2,4,5], [3,5,1,7])` の計算を机上で行い、その簡約列を示せ。

解答例. 以下の通り.

```
union([2,4,5], [3,5,1,7])  
= 2:union([4,5], [3,5,1,7])  
= 2:4:union([5], [3,5,1,7])  
= 2:4:union([], [3,5,1,7])  
= 2:4:[3,5,1,7]  
= [2,4,3,5,1,7]
```

□

問題 3 (B 問題). 2 つの集合の差集合を求める関数 `diff` を定義せよ. また, 関数 `diff` を用いて下記の差集合を求め, その結果を示せ.

- 空集合から集合 $\{3, 5, 1, 7\}$ を除いた差集合
- 集合 $\{3, 5, 1, 7\}$ から空集合を除いた差集合
- 集合 $\{2, 4, 5\}$ から集合 $\{3, 5, 1, 7\}$ を除いた差集合

解答例. 以下の通り.

```
diff([], bs) = []  
diff(a:as, bs) = if memberOf(bs, a) then diff(as, bs)  
                 else a:diff(as, bs)
```

-- 実行結果 --

```
diff([], [3,5,1,7]) = []  
diff([3,5,1,7], []) = [3,5,1,7]  
diff([2,4,5], [3,5,1,7]) = [2,4]
```

□

問題 4 (B 問題). `diff([2,4,5], [3,5,1,7])` の計算を机上で行い, その簡約列を示せ.

解答例. 以下の通り.

```
diff([2,4,5], [3,5,1,7])
= 2:diff([4,5], [3,5,1,7])
= 2:4:diff([5], [3,5,1,7])
= 2:4:diff([], [3,5,1,7])
= 2:4:[]
= [2,4]
```

□

問題 5 (B 問題). 2 つの集合の積集合を求める関数 `inter` を定義せよ. また, 関数 `inter` を用いて下記の積集合を求め, その結果を示せ.

- 空集合と集合 $\{3, 5, 1, 7\}$ の積集合
- 集合 $\{3, 5, 1, 7\}$ と空集合の積集合
- 集合 $\{1, 2, 3, 4, 5\}$ と集合 $\{3, 5, 1, 7\}$ の積集合

解答例. 以下の通り.

```
inter([], bs) = []
inter(a:as, bs) = if memberOf(bs, a) then a:inter(as, bs)
                  else inter(as, bs)

-- 実行結果 --
inter([], [3,5,1,7]) = []
inter([3,5,1,7], []) = []
inter([1,2,3,4,5], [3,5,1,7]) = [1,3,5]
```

□

問題 6 (B 問題). `inter([1,2,3,4,5], [3,5,1,7])` の計算を机上で行い, その簡約列を示せ.

解答例. 以下の通り.

```
inter([1,2,3,4,5], [3,5,1,7])
= 1:inter([2,3,4,5], [3,5,1,7])
= 1:inter([3,4,5], [3,5,1,7])
= 1:3:inter([4,5], [3,5,1,7])
= 1:3:inter([5], [3,5,1,7])
= 1:3:5:inter([], [3,5,1,7])
= 1:3:5:[]
= [1,3,5]
```

□

問題 7 (B 問題). 集合 A が集合 B の部分集合であれば True を, そうでなければ False を返す関数 `subset` を定義せよ. また, 関数 `subset` を用いて下記の評価を行い, その結果を示せ.

- 空集合が集合 {3, 5, 1, 7} の部分集合か評価
- 集合 {3, 5, 1, 7} が 空集合の部分集合か評価
- 集合 {1, 3} が 集合 {3, 5, 1, 7} の部分集合か評価
- 集合 {2, 3} が 集合 {3, 5, 1, 7} の部分集合か評価

解答例. 以下の通り.

```
subset([], bs) = True
subset(a:as, bs) = if memberOf(bs, a) then subset(as, bs)
                  else False

-- 実行結果 --
subset([], [3,5,1,7]) = True
subset([3,5,1,7], []) = False
subset([1,3], [3,5,1,7]) = True
subset([2,3], [3,5,1,7]) = False
```

□

問題 8 (B 問題). `subset([1,3], [3,5,1,7])` の計算を机上で行い, その簡約列を示せ.

解答例. 以下の通り.

```
subset([1,3], [3,5,1,7])  
= subset([3], [3,5,1,7])  
= subset([], [3,5,1,7])  
= True
```

□

問題 9 (B 問題). 2 つの集合が等しければ `True` を, そうでなければ `False` を返す関数 `equalSet` を定義せよ. また, 関数 `equalSet` を用いて下記の評価を行い, その結果を示せ.

- 空集合が空集合と等しいか評価
- 集合 $\{3, 2, 6\}$ が 集合 $\{2, 6, 3\}$ と等しいか評価
- 集合 $\{3, 2, 6\}$ が 集合 $\{2, 6, 4\}$ と等しいか評価

解答例. 以下の通り.

```
equalSet(as, bs) = if subset(as, bs) && subset(bs, as)  
                  then True else False  
  
-- 実行結果 --  
equalSet([], []) = True  
equalSet([3,2,6], [2,6,3]) = True  
equalSet([3,2,6], [2,6,4]) = False
```

□

問題 10 (B 問題). 集合の要素を 2 倍にした集合を求める関数 `doubleSet` を定義せよ. また, 関数 `doubleSet` を用いて下記の集合を評価した結果を示せ.

- 集合 $\{2, 5, 1\}$
- 集合 $\{3, 1, 4, 6, 2\}$

解答例. 以下の通り.

```
doubleSet([]) = []  
doubleSet(a:as) = a*2:doubleSet(as)  
  
別解:  
doubleSet([]) = []  
doubleSet(a:as) = a+a:doubleSet(as)  
  
-- 実行結果 --  
doubleSet([2,5,1]) = [4,10,2]  
doubleSet([3,1,4,6,2]) = [6,2,8,12,4]
```

□

問題 11 (B 問題). 集合の要素を x 倍にした集合を求める関数 `multipleSet` を定義せよ. また, 関数 `multipleSet` を用いて下記の評価を行い, その結果を示せ.

- 集合 $\{2, 5, 1\}$ の要素を 3 倍に
- 集合 $\{3, 1, 4, 6, 2\}$ の要素を 10 倍に

解答例. 以下の通り.

```
multipleSet([], x) = []  
multipleSet(a:as, x) = a*x:multipleSet(as, x)  
  
-- 実行結果 --  
multipleSet([2,5,1], 3) = [6,15,3]  
multipleSet([3,1,4,6,2], 10) = [30,10,40,60,20]
```

□

問題 12 (A 問題). 集合の要素のうち, 偶数だけを残した集合を求める関数 `evenSet` を定義せよ. また, 関数 `evenSet` を用いて下記の集合を評価した結果を示せ.

- 集合 $\{0, 1, 2, 3, 4, 5, 6\}$
- 集合 $\{0, 2, 4, 6, 8, 10\}$
- 集合 $\{1, 3, 5, 7, 9, 11\}$

解答例. 以下の通り.

```
evenSet([]) = []
evenSet(a:as) = if a 'mod' 2 == 0 then a:evenSet(as)
                else evenSet(as)

-- 実行結果 --
evenSet([0,1,2,3,4,5,6]) = [0,2,4,6]
evenSet([0,2,4,6,8,10]) = [0,2,4,6,8,10]
evenSet([1,3,5,7,9,11]) = []
```

□

問題 13 (B 問題). 集合の要素のうち, 奇数だけを残した集合を求める関数 `oddSet` を定義せよ. また, 関数 `oddSet` を用いて下記の集合を評価した結果を示せ.

- 集合 $\{0, 1, 2, 3, 4, 5, 6\}$
- 集合 $\{0, 2, 4, 6, 8, 10\}$
- 集合 $\{1, 3, 5, 7, 9, 11\}$

解答例. 以下の通り.

```
oddSet([]) = []
oddSet(a:as) = if a 'mod' 2 == 1 then a:oddSet(as)
                else oddSet(as)

別解:
oddSet([]) = []
oddSet(a:as) = if a 'mod' 2 == 0 then oddSet(as)
                else a:oddSet(as)

-- 実行結果 --
oddSet([0,1,2,3,4,5,6]) = [1,3,5]
oddSet([0,2,4,6,8,10]) = []
oddSet([1,3,5,7,9,11]) = [1,3,5,7,9,11]
```

□

問題 14 (B 問題). 集合の要素を 2 乗にした集合を求める関数 `squareSet` を定義せよ. また, 関数 `squareSet` を用いて下記の集合を評価した結果を示せ.

- 集合 $\{2, 5, 1\}$
- 集合 $\{3, 1, 4, 6, 2\}$

解答例. 以下の通り.

```
squareSet([]) = []  
squareSet(a:as) = (a*a):squareSet(as)  
  
-- 実行結果 --  
squareSet([2,5,1]) = [4,25,1]  
squareSet([3,1,4,6,2]) = [9,1,16,36,4]
```

□

問題 15 (B 問題). 集合の要素のうち, 正の値だけを残した集合を求める関数 `positiveSet` を定義せよ. また, 関数 `positiveSet` を用いて下記の集合を評価した結果を示せ.

- 集合 $\{0, 1, 2, 3, 4, 5\}$
- 集合 $\{-4, -3, -2, -1, 0\}$
- 集合 $\{-3, -1, 0, 1, 5, -2, 8\}$

解答例. 以下の通り.

```
positiveSet([]) = []  
positiveSet(a:as) = if a > 0 then a:positiveSet(as)  
                   else positiveSet(as)  
  
-- 実行結果 --  
positiveSet([0,1,2,3,4,5]) = [1,2,3,4,5]  
positiveSet([-4,-3,-2,-1,0]) = []  
positiveSet([-3,-1,0,1,5,-2,8]) = [1,5,8]
```

□

問題 16 (B 問題). 集合の要素のうち, x より大きい値だけを残した集合を求める関数 `greaterThanSet` を定義せよ. また, 関数 `greaterThanSet` を用いて下記の評価を行い, その結果を示せ.

- 集合 $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ から 60 より大きい値の集合を求める
- 集合 $\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ から 60 より大きい値の集合を求める

解答例. 以下の通り.

```
greaterThanSet([], x) = []
greaterThanSet(a:as, x) = if a > x then a:greaterThanSet(as, x)
                           else greaterThanSet(as, x)

-- 実行結果 --
greaterThanSet([10,20,30,40,50,60,70,80,90,100], 60) = [70,80,90,100]
greaterThanSet([75,54,98,72,45,60,88,59,92,55,35], 60) = [75,98,72,88,92]
```

□

問題 17 (B 問題). 集合の要素のうち, x 以上 y 以下の値だけを残した集合を求める関数 `rangeSet` を定義せよ. また, 関数 `rangeSet` を用いて下記の評価を行い, その結果を示せ.

- 集合 $\{1, 2, 3, 4, 5\}$ のうち 2 以上 4 以下の要素の集合を求める
- 集合 $\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ のうち 60 以上 80 以下の要素の集合を求める

解答例. 以下の通り.

```
rangeSet([], x, y) = []
rangeSet(a:as, x, y) = if x <= a && a <= y then a:rangeSet(as, x, y)
                       else rangeSet(as, x, y)

-- 実行結果 --
rangeSet([1,2,3,4,5], 2, 4) = [2,3,4]
rangeSet([75,54,98,72,45,60,88,59,92,55,35], 60, 80) = [75,72,60]
```

□

問題 18 (B 問題). 実習資料 第 03-04 時限の問題 48 で作成した関数 `countRange` を, 関数 `rangeSet` と関数 `numberOf` の合成関数 `countRangeB` として実現せよ. また, 関数 `countRangeB` を用いて下記の評価を行い, その結果を示せ.

- 集合 $\{1, 2, 3, 4, 5\}$ のうち 2 以上 4 以下の要素の個数を求める
- 集合 $\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ のうち 60 以上 80 以下の要素の個数を求める

解答例. 以下の通り.

```
countRangeB(as, x, y) = numberOf(rangeSet(as, x, y))

countRangeB([1,2,3,4,5], 2, 4) = 3
countRangeB([75,54,98,72,45,60,88,59,92,55,35], 60, 80) = 3
```

□

問題 19 (B 問題). 集合の要素のうち, x 以上 y 以下の値の合計を求める関数 `sumRange` を定義せよ. また, 関数 `sumRange` を用いて下記の評価を行い, その結果を示せ.

- 集合 $\{1, 2, 3, 4, 5\}$ のうち 2 以上 4 以下の要素の合計を求める
- 集合 $\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ のうち 60 以上 80 以下の要素の合計を求める

解答例. 以下の通り.

```
sumRange(as, x, y) = sumSet(rangeSet(as, x, y))

-- 実行結果 --
sumRange([1,2,3,4,5], 2, 4) = 9
sumRange([75,54,98,72,45,60,88,59,92,55,35], 60, 80) = 207
```

□

問題 20 (D 問題). 値 n が素数ならば `True` を, 素数でなければ `False` を返す関数 `prime` を定義せよ. また, 関数 `prime` を用いて n を 1 から 8 まで順に評価した結果をそれぞれ示せ.

n を 2 から n まで順に割って, 最後まで割り切れなければ素数であると判定できる (最も単純な判定方法であり, 無駄が多いので大きな値の判定には向かない). 補助関数として, n と i が等しくなれば `True` を返し, そうでない場合, n が i で割り切れれば `False` を返し, 割り切れなければ $i+1$ に対して再帰処理を行う関数 `prime0(n, i)` を作成し, 関数 `prime` は n が 2 未満のとき `False` を返し, それ以外のとき `prime0` を $i=2$ として呼び出すようにするとよい.

解答例. 以下の通り.

```
prime0(n, i) = if n == i then True
               else if n 'mod' i == 0 then False
               else prime0(n, i+1)

prime(n) = if n < 2 then False
           else prime0(n, 2)

-- 実行結果 --
prime(1) = False
prime(2) = True
prime(3) = True
prime(4) = False
prime(5) = True
prime(6) = False
prime(7) = True
prime(8) = False
```

□

問題 21 (D 問題). 値 n 以下の素数の集合を求める関数 `primeSet` を, 関数 `prime` を用いて定義せよ. また, 関数 `primeSet` を用いて 100 以下の素数の集合を求め, その結果を示せ

解答例. 以下の通り.

```
primeSet0(n, i) = if i > n then []  
                  else if prime(i) then i:primeSet0(n, i+1)  
                  else primeSet0(n, i+1)
```

```
primeSet(n) = if n < 1 then []  
              else primeSet0(n, 1)
```

別解: 降順になる

```
primeSetB(n) = if n < 1 then []  
               else if prime(n) then n:primeSetB(n-1)  
               else primeSetB(n-1)
```

-- 実行結果 --

```
primeSet(100) = [2,3,5,7,11,13,17,19,23,29,31,37,41,  
                 43,47,53,59,61,67,71,73,79,83,89,97]  
primeSetB(100) = [97,89,83,79,73,71,67,61,59,53,47,43,  
                  41,37,31,29,23,19,17,13,11,7,5,3,2]
```

□

問題 22 (B 問題). リストの要素のうち, 最小値の要素の値を求める関数 `minList` を定義せよ. また, 定義した関数を用いて, 下記の整数列に適用した結果を示せ.

- リスト `[4, 3, 7]`
- リスト `[4]`
- リスト `[1, 2, 3]`
- リスト `[3, 2, 1]`
- リスト `[2, 2, 1, 1]`

解答例. 以下の通り.

```
minList([a]) = a
minList(a:as) = if a < minList(as) then a else minList(as)
```

実行例:

```
*Main> minList([4,3,7])
3
*Main> minList([4])
4
*Main> minList([1,2,3])
1
*Main> minList([3,2,1])
1
*Main> minList([2,2,1,1])
1
```

□

問題 23 (B 問題). リストから, 指定した値の要素を取り除く関数 `minus` を定義せよ. 複数回出現する場合は, 先頭に近い要素のみを取り除く. また, 定義した関数を用いて, 下記の整数列に適用した結果を示せ.

- リスト `[4, 3, 7]` から 3
- リスト `[4]` から 4
- リスト `[1, 2, 3]` から 4
- リスト `[3, 2, 1]` から 2
- リスト `[2, 2, 1, 1]` から 2

解答例. 以下の通り.

```
minus([], b) = []  
minus(a:as,b) = if a == b then as else a : minus(as,b)
```

実行例:

```
*Main> minus([4,3,7], 3)  
[4,7]  
*Main> minus([4], 4)  
[]  
*Main> minus([1,2,3], 4)  
[1,2,3]  
*Main> minus([3,2,1], 2)  
[3,1]  
*Main> minus([2,2,1,1], 2)  
[2,1,1]
```

□

問題 24 (B 問題). 整数列を昇順に並べ替えるソート関数 `sort` を定義せよ. また, 定義した関数を用いて, 下記の整数列を並べ替えた結果を示せ.

- リスト [4, 3, 7]
- リスト [4]
- リスト [1, 2, 3]
- リスト [3, 2, 1]
- リスト [2, 2, 1, 1]

解答例. 以下の通り.

```
sort([]) = []  
sort([x]) = [x]  
sort(a:as) = if a < head(sort(as)) then a:sort(as)  
             else head(sort(as)):sort(a:tail(sort(as)))
```

実行例:

```
*Main> sort([4,3,7])  
[3,4,7]  
*Main> sort([4])  
[4]  
*Main> sort([1,2,3])  
[1,2,3]  
*Main> sort([3,2,1])  
[1,2,3]  
*Main> sort([2,2,1,1])  
[1,1,2,2]
```

□

問題 25 (B 問題). 整数列を降順に並べ替えるソート関数 `dsort` を定義せよ. また, 定義した関数を用いて, 下記の整数列を並べ替えた結果を示せ.

- リスト [4, 3, 7]
- リスト [4]
- リスト [1, 2, 3]
- リスト [3, 2, 1]
- リスト [5, 6, 5, 6]

解答例. 以下の通り.

```
dsort([]) = []  
dsort([x]) = [x]  
dsort(a:as) = if a > head(dsor(as)) then a:dsort(as)  
              else head(dsor(as)):dsort(a:tail(dsor(as)))
```

実行結果:

```
*Main> dsort([4,3,7])  
[7,4,3]  
*Main> dsort([4])  
[4]  
*Main> dsort([1,2,3])  
[3,2,1]  
*Main> dsort([3,2,1])  
[3,2,1]  
*Main> dsort([5,6,5,6])  
[6,6,5,5]
```

□

問題 26 (参考). 同じ値やリストを用いて何度か関数を評価させるときは, その値やリストを変数で参照するようにすると, 入力が楽になる. 変数に値を入れるには `let` を用いて次のように記述する.

```
*Main> let d = [3, 8, 4, 5, 2]
```

このリストを並べ替えるときは次のようにリストの代わりに変数名を指定した式を書けばよい.

```
*Main> sort(d)  
[2,3,4,5,8]
```

また, 変数 `d` がどのような値やリストになっているかは, 次のように名前だけの式を書けば確認できる.

```
*Main> d  
[3,8,4,5,2]
```

関数 `sort` は, リストの並べ替えを行う関数であるが, `sort(d)` を実行したときに, その前後で変数 `d` が参照するリストが変化するかどうかを, 実際に実行して確認せよ. また, その結果, どちらであったか答えよ.

解答例以下の通り.

```
let d = [3, 8, 4, 5, 2]
*Main> d
[3,8,4,5,2]
*Main> sort(d)
[2,3,4,5,8]
*Main> d
[3,8,4,5,2]
```

答え: 変化しない



問題 27 (参考). 以下の関数 `mkdata100` で, 0 以上 100 未満の整数値で構成される乱数を m 個作ることができる.

```
mkdata100(m, s) = [rand100(n, s) | n <- [0..(m-1)]]
rand100(n, s) = if n==0 then s 'mod' 100 else (21*rand100(n-1, s)+13) 'mod' 100
```

ここで, s は乱数を発生する種であり, 適当な整数値を指定するものとする. s の値を変えると, 他の乱数列を作ることができる.

この関数 `mkdata100` を定義し, 10 個の乱数を生成させた結果を示せ. ただし, 種として用いる整数値は, 各自の学生番号の 3 桁の数字 (14SE987 なら 987) とし, 以降の問題でもすべて同じ種を用いること.

解答例. `mkdata100(10, 987)` の場合:

```
[87,40,53,26,59,52,5,18,91,24]
```



問題 28 (参考). 関数 `mkdata100` を使って生成した 10 個の乱数列を昇順に並べ替える式を関数合成を用いて記述せよ.

解答例. 以下の通り.

```
sort(mkdata100(10,987))
```

3 秒以上かかる個数: 23 個 (環境に依存)



問題 29 (B 問題). 整数列が昇順に並んでいるかを判定し, 並んでいる場合は `True` を, 並んでいない場合は `False` を返す関数 `isSorted` を定義せよ. 昇順に並んでいることは, リストを先頭の要素 a と残りのリスト

as に分けたときに, as が並んでいて, かつ, as の先頭の要素が a 以上になることで判定できる. なお, 先頭の要素は関数 head で取り出すことができるが, 空のリストを与えるとエラーになることに注意せよ.

また, 次のリストを与えたときの結果を示せ.

- リスト [1, 2, 4, 8]
- リスト [8, 4, 2, 1]
- リスト [2, 2, 2, 2]

解答例. 以下の通り.

```
isSorted([]) = True
isSorted([a]) = True
isSorted(a:as) = isSorted(as) && a <= head(as)
```

□

問題 30 (参考). 関数 mkdata100 を用いて 20 個の乱数のリストを作り, それが並んでいるかを関数 isSorted で判定する式を関数合成を用いて記述せよ. また, その実行結果も示せ.

同様に, 乱数のリストを関数 sort で並べ替えてから判定する式も記述し, その実行結果も示せ.

解答例. 以下の通り.

```
isSorted(mkdata100(20,987))
False

isSorted(sort(mkdata100(20,987)))
True
```

□

問題 31 (B 問題). 2 つのリストを連結して一つのリストを返す関数 append を定義せよ. 例えば, append([1, 8, 3], [6, 2]) はリスト [1, 8, 3, 6, 2] を返す. 基本的には, 2 つのうち, 最初の方のリストを先頭の要素 a と残りの要素 as に分け, as ともう一方のリストを連結して得られたリストを求め, その先頭に a を加えたリストを構成すればよい.

解答例. 以下の通り.

```
append([], bs) = bs
append(a:as, bs) = a:append(as, bs)
```

□

問題 32 (B 問題). 整数リストを逆順に並べる関数 `reverseList` を定義せよ. 基本的には, リストを先頭の要素と残りのリストに分けたときに, 残りのリストを逆順に並べたものに, 先頭の要素だけを持つリストを連結すればよい. リストの連結には関数 `append` を用いること.

解答例. 以下の通り.

```
reverseList([]) = []  
reverseList(a:as) = append(reverseList(as), [a])
```

□

問題 33 (B 問題). 整数リストを降順に並べ替えるソート関数 `sortR` を, 関数 `sort` と `reverseList` を用いて定義せよ.

解答例. 以下の通り.

```
sortR(as)=reverse(sort(as))
```

□

問題 34 (B 問題). リストを集合の表現として用いるためには, そのリストには重複した値が入っていないことが前提となる. リストが集合として使えるかどうかを調べるために, リストの中に重複した値がない場合には `True` を, そうでない場合は `False` を返す関数 `hasOnlyUniqueNum` を定義しなさい. 例えば, リスト `[3, 1, 6, 5, 7]` に対しては `True` を, `[1, 4, 2, 8, 4, 5]` に対しては `False` を返す. なお, リストの中にある値が含まれるかどうかを判定するためには, 関数 `memberOf` を定義して使用するとよい.

解答例. 以下の通り.

```
hasOnlyUniqueNum([]) = True  
hasOnlyUniqueNum(a:as) = if memberOf(as, a) then False  
                           else hasOnlyUniqueNum(as)
```

□

問題 35 (B 問題). リストの中に重複する値が含まれる場合に, その重複した値を削除したリストを返す関数 `toUnique` を定義しなさい. 例えば, リスト `[1, 4, 2, 8, 4, 2, 3]` に対しては `[1, 8, 4, 2, 3]` を返す.

解答例. 以下の通り.

```
toUnique([]) = []  
toUnique(a:as) = if memberOf(as, a) then toUnique(as) else a:toUnique(as)
```

□

問題 36 (B 問題). 昇順に並んだ 2 つの整数列を 1 つの整数列に合成する関数 `mergeSortedList` を再帰的に定義せよ. 例えば, リスト `[1, 3, 5]` と `[2, 4, 6]` 合成すると `[1, 2, 3, 4, 5, 6]` になる. 合成する際には, 2 つのリストの先頭の要素のうちどちらが小さいかを判定し, 小さい方の後ろに残りの整数列を合成したものを結合すればよい. 例えば, リスト `[1, 3, 5]` と `[2, 4, 6]` 合成するときは, 先頭の要素の 1 と 2 のうち, 小さい方は 1 であるので, 1 の後ろに `[3, 5]` と `[2, 4, 6]` を合成したリストを結合させる.

解答例. 以下の通り.

```
mergeSortedList([], bs) = bs  
mergeSortedList(as, []) = as  
mergeSortedList(a:as, b:bs) = if a < b then a:mergeSortedList(as, b:bs)  
                               else b:mergeSortedList(a:as, bs)
```

□

問題 37 (B 問題). 昇順に並んだ 2 つの整数列から共通する要素のリストを求める関数 `interSortedList` を再帰的に定義せよ.

解答例. 以下の通り.

```
interSortedList([], bs) = []  
interSortedList(as, []) = []  
interSortedList(a:as, b:bs) = if a < b then interSortedList(as, b:bs)  
                               else if a > b then interSortedList(a:as, bs)  
                               else a:interSortedList(as, bs)
```

□

問題 38 (B 問題). 整数が任意の順序に並んだ 2 つの整数列から共通する要素のリストを求める関数 `interSortedList2` を, 関数 `sort` と `interSortedList` を用いて定義せよ.

解答例. 以下の通り.

```
interSortedList2(as, bs) = interSortedList(sort(as), sort(bs))
```

□

問題 39 (B 問題). 昇順に並んでいる整数リストに, 昇順を維持したまま新しい整数を挿入する関数 `insertSortedList(a:as, n)` を定義せよ. 例えば, 整数リスト `[2, 4, 8]` に 7 を挿入すると, リストは `[2, 4, 7, 8]` になる. 基本的には, リスト `a:as` に値 `n` を挿入する場合, `n` が `a` 以下のときは, `n` はリストのどの値よりも小さいか等しいので, 先頭に挿入する. そうでない場合は, リストの残りの `as` の中に挿入する.

解答例. 以下の通り.

```
insertSortedList([], n) = [n]
insertSortedList(a:as, n) = if a >= n then n:a:as
                             else a:insertSortedList(as, n)
```

□

問題 40 (B 問題). 関数 `insertSortedList` を用いたソート関数 `insSort` を定義せよ. 基本的には, 並べ替え対象のリストを先頭の要素と残りの要素のリストに分けた場合, 残りの要素のリストを並べ替えたものに先頭の要素を挿入すればよい.

解答例. 以下の通り.

```
insSort([]) = []
insSort(a:as) = insertSortedList(insSort(as), a)
```

□