プログラミング基礎 実習資料 第 03-04 時限

問題 1 (B 問題)**.** 次の式をセッションで評価しなさい．評価結果が自分の考えた結果と同じか確認しなさい．

(1) `5 / 2`
(2) `` 5 `div` 2 ``
(3) `` 5 `mod` 2 ``
(4) `33 / 7`
(5) `` 33 `div` 7 ``
(6) `` 33 `mod` 7 ``
(7) `100 / 5`
(8) `` 100 `div` 5 ``
(9) `` 100 `mod` 5 ``
(10) `` 1234 `div` 10 ``
(11) `` 1234 `mod` 10 ``
(12) `20 / 5 * 2`
(13) `20 / (5 * 2)`

解答例. 以下の通り．

(1) `2.5`
(2) `2`
(3) `1`
(4) `4.714285714285714`
(5) `4`
(6) `5`
(7) `20.0`
(8) `20`
(9) `0`
(10) `123`
(11) `4`
(12) `8.0`
(13) `2.0`

□

問題 2 (B 問題)．階乗を計算する関数 `fact` を Haskell で定義せよ．

解答例．以下の通り．

```
fact(0) = 1
fact(n) = n * fact(n-1)
```

□

問題 3 (B 問題)．順列の数は階乗を用いて

$$ {}_n\mathrm{P}_k = \frac{n!}{(n-k)!} $$

と表せる．この定義にしたがい，Haskell で順列を計算する関数 `permA(n,k)` を定義せよ．

解答例．以下の通り．

```
permA(n, k) = fact(n) `div` fact(n-k)
```

□

問題 4 (B 問題)．組合せの数は順列と階乗を用いて

$$ {}_n\mathrm{C}_k = \frac{{}_n\mathrm{P}_k}{k!} $$

と表せる．この定義にしたがい，Haskell で組合せを計算する関数 `combA(n,k)` を定義せよ．

解答例．以下の通り．

```
combA(n, k) = permA(n, k) `div` fact(k)
```

□

問題 5 (A 問題)．5 人から 3 人を選び出して並べる方法は何通りか．また，5 人の中から 3 人を単に選び出す方法は何通りか．

解答例．5 人から 3 人を選び出して並べる方法は，$60(= {}_5\mathrm{P}_3)$ 通りである．5 人から 3 人を単に選び出す方法は，$10(= {}_5\mathrm{C}_3)$ 通りである． □

問題 6 (B 問題)．10 人から 5 人を選び出して並べる方法は何通りか．また，10 人の中から単に 5 人を選び出す方法は何通りか．

解答例．10 人から 5 人を選び出して並べる方法は 30240(= ${}_{10}\mathrm{P}_5$) 通りである．10 人の中から単に 5 人を選び出す方法は，252(= ${}_{10}\mathrm{C}_5$) 通りである． □

問題 7 (B 問題)．順列の数は次のように再帰的定義で表すこともできる．

$$\begin{aligned} {}_n\mathrm{P}_0 &= 1 \\ {}_n\mathrm{P}_k &= n \cdot {}_{n-1}\mathrm{P}_{k-1} \qquad \textbf{if } 0 < k \le n \end{aligned} \tag{1}$$

この定義にしたがい，Haskell で順列の数を計算する関数 `permB(n,k)` を定義せよ．

解答例．以下の通り．

```
permB(n, 0) = 1
permB(n, k) = n * permB(n-1, k-1)
```

□

問題 8 (B 問題)．組合せの数は次のように再帰的定義で表すこともできる．

$$\begin{aligned} {}_n\mathrm{C}_0 &= 1 \\ {}_n\mathrm{C}_n &= 1 \\ {}_n\mathrm{C}_k &= {}_{n-1}\mathrm{C}_k + {}_{n-1}\mathrm{C}_{k-1} \qquad \textbf{if } 1 \le k \le n-1 \end{aligned} \tag{2}$$

この定義にしたがい，Haskell で組合せの数を計算する関数 `combB(n,k)` を定義せよ．

解答例．以下の通り．

```
combB(n, 0) = 1
combB(n, k) = if n==k then 1
              else combB(n-1, k) + combB(n-1, k-1)
```

□

問題 9 (B 問題)．組合せの数は階乗を用いて

$$ {}_n\mathrm{C}_k = \frac{n!}{(n-k)! \cdot k!} $$

とも表せる．この定義にしたがい，Haskell で組合せの数を計算する関数 `combA2(n,k)` を定義せよ．

解答例．以下の通り．

```
combA2(n, k) = fact(n) `div` (fact(n-k)*fact(k))
```

□

問題 10 (A 問題)．n の階乗を if による場合分けを用いて，関数 `fact2` として作成せよ．

解答例. 以下の通り.

```
fact2(n) = if n==0 then 1
           else n * fact2(n-1)
```

□

問題 11 (B 問題). 実習資料 第 02 時限の問題 15 の 0 から非負整数 n までの整数の総和を計算する関数 `mysum` を，if による場合分けを用いて，関数 `mysum2` として作成せよ.

解答例. 以下の通り.

```
mysum2(n) = if n==0 then 0
            else n + mysum2(n-1)
```

□

問題 12 (B 問題). 実習資料 第 02 時限の問題 17 の 0 から非負整数 n までの整数の 2 乗和を計算する関数 `sumsq` を，if による場合分けを用いて，関数 `sumsq3` として作成せよ.

解答例. 以下の通り.

```
sumsq3(n) = if n==0 then 0
            else n*n + sumsq3(n-1)
```

□

問題 13 (B 問題). 実習資料 第 02 時限の問題 18 フィボナッチ数列を計算する関数 `fib` を，if による場合分けを用いて，関数 `fib2` として作成せよ.

解答例. 以下の通り.

```
fib2(n) = if n==0 then 0
          else if n==1 then 1
          else fib2(n-1) + fib2(n-2)
```

□

問題 14 (A 問題). ニュートン法により $\sqrt{2}$ を計算する関数 `root2(n)` を定義せよ.

解答例．$f(x)=0$ の方程式の 1 つの近似解をニュートン法で求めるための漸化式は次のようになる．

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$\sqrt{2}$ は方程式 $f(x)=x^2-2=0$ の解の 1 つで，$f'(x)=2x$ より，近似解を求めるための漸化式は

$$x_{n+1} = x_n - \frac{x_n^2-2}{2x_n} = \frac{x_n}{2} + \frac{1}{x_n}$$

である．

初期値を $x_0=2$ として，これを Haskell の関数 `root2` として定義すると次のようになる．

```
root2(0) = 2
root2(n) = root2(n-1)/2 + 1/root2(n-1)
```

□

問題 15 (B 問題)**.** 問題 14 では，ニュートン法において，`n` が `0` のときの初期値を `2` としている．初期値を `128`, `0.1`, `0`, `-2` とした場合，それぞれ，それ以降の項の値はどのように変化するかを考えよ．問題 14 の `root2(0)` の値を変更して実際に計算し，想定した結果となるか，確認せよ．

解答例．以下の通り．

- $x_0=128$ とすると，最初は x_n は $\sqrt{2}$ から大きく離れた値となるが，繰り返すにつれ，x_n は $\sqrt{2}$ に近づく (収束する)．
- $x_0=0.1$ とすると，x_1 は解から離れるが，その後，$\sqrt{2}$ に収束する．
- $x_0=0$ とすると，接線が x 軸と交わらないので，$x_n (n>0)$ は得られない．
- $x_0=-2$ とすると，$-\sqrt{2}$ に収束する．

□

問題 16 (B 問題)**.** ニュートン法により $\sqrt{3}$ を計算する関数 `root3(n)` を定義せよ．ただし，`n` が `0` のときの初期値を `3` とする．実際に `root3(1)`〜`root3(6)` まで計算し，徐々に $\sqrt{3}=1.7320508075688772935 2...$ に近づくことを確認せよ．

解答例．以下の通り．

```
root3(0) = 3
root3(n) = root3(n-1)/2 + 3/(2*root3(n-1))

> root3(1)
2.0
> root3(2)
1.75
> root3(3)
1.7321428571428572
> root3(4)
1.7320508100147274
> root3(5)
1.7320508075688772
> root3(6)
1.7320508075688772
```

□

問題 17 (B 問題)．100 点満点の点数 (Score) を 5 段階評価の点数 (Grade Point) に変換する関数 `s2gp` を，if を用いて作成せよ．点数の関係は次の表の通りである．なお，Score が 100 を超える場合やマイナスの場合は，エラーとして 5 段階評価の点数を -1 とする．(ヒント: else と if を適切に組み合わせるとよい)

100 点満点 (Score)	5 段階評価 (Grade Point)
101〜	-1
90〜100	4
80〜89	3
70〜79	2
60〜69	1
0〜59	0
〜-1	-1

解答例. 以下の通り.

```
s2gp(x) = if x>100 then -1
          else if x>=90 then 4
          else if x>=80 then 3
          else if x>=70 then 2
          else if x>=60 then 1
          else if x>=0 then 0
          else -1
```

□

問題 18 (B 問題)**.** 正の整数の 1 の位を求める関数 `keta1` を定義せよ（ヒント: 余りの計算を利用する).

解答例. 以下の通り.

```
keta1(n) = n `mod` 10
```

□

問題 19 (B 問題)**.** 正の整数の 10 の位より上位の桁からなる整数を求める関数 `keta10` を定義せよ．例えば，`keta10(1234)` は 123 に，`keta10(864200)` は 86420 に，`keta10(8)` は 0 になる．（ヒント: 商の計算を利用する).

解答例. 以下の通り.

```
keta10ijou(n) = n `div` 10
```

□

問題 20 (B 問題)**.** 正の整数の各桁の合計を求める関数 `keta_goukei` を定義せよ．関数 `keta1` や `keta10` を使ってもよい．例えば，`keta_goukei(1234)` は 10 に，`keta_goukei(864200)` は 20 に，`keta_goukei(8)` は 8 になる．

なお，正の整数 n の各桁の合計を次のようにして再帰的に計算できる．

- n が 1 桁の整数ならば，桁の合計は n
- そうでなければ，桁の合計は (n の 1 の位の数 + n の 10 の位より上位の桁からなる整数の桁の合計)

解答例. 以下の通り.

```
keta_goukei(n) = if n<10 then n
                 else keta1(n) + keta_goukei(keta10(n))
```

□

問題 21 (B 問題). 生年月日による次のような相性占いがある.

1. 2 人のそれぞれの生年月日の各桁の数をすべて足す.
2. 1. で求めた 2 つの数の大きい方から小さい方を引く.
3. 2. で求めた数を 5 で割った余りが大きいほど相性がよい. つまり, 4 が最高で, 0 は最低である.

生年月日を 8 桁の整数で表したとき, 2 人の生年月日から相性を計算する関数 `aishou` を関数 `keta_goukei`, `diff`(実習資料 第 02 時限の問題 21) の合成関数を用いて作成せよ. 例えば, 1993 年 10 月 28 日生まれの兄と 1996 年 9 月 17 日生まれの弟の相性は `aishou(19931028, 19960917)` で計算でき, 結果は 4(最高) である.

解答例. 以下の通り.

```
aishou(a, b) = diff(keta_goukei(a), keta_goukei(b)) `mod` 5
```

□

問題 22 (B 問題). 実習資料 第 02 時限の問題 32 のリュカ数列を計算する関数 `lucas` を, if による場合分けを用いて, 関数 `lucas2` として作成せよ.

解答例. 以下の通り.

```
lucas2(n) = if n==0 then 2
            else if n==1 then 1
            else lucas2(n-1) + lucas2(n-2)
```

□

問題 23 (B 問題). 実習資料 第 02 時限の問題 33 のトリボナッチ数列を計算する関数 `tribo` を, if による場合分けを用いて, 関数 `tribo2` として作成せよ.

解答例. 以下の通り.

```
tribo2(n) = if n==0 then 0
            else if n==1 then 1
            else if n==2 then 1
            else tribo2(n-1) + tribo2(n-2) + tribo2(n-3)
```

□

問題 24 (B 問題)**.** 実習資料 第 02 時限の問題 35 の整数の 3 乗の和を求める関数 sumcb を，if による場合分けを用いて，関数 sumcb2 として作成せよ.

解答例. 以下の通り.

```
sumcb2(n) = if n==1 then 1 else n^3 + sumcb2(n-1)
```

□

問題 25 (B 問題)**.** 実習資料 第 02 時限の問題 36 の 2 から n 番目までの偶数の 2 乗の和を求める関数 sumsqeven を，if による場合分けを用いて，関数 sumsqeven2 として作成せよ.

解答例. 以下の通り.

```
sumsqeven2(n) = if n==1 then 4 else (2*n)^2 + sumsqeven2(n-1)
```

□

問題 26 (B 問題)**.** 実習資料 第 02 時限の問題 37 の 1 から n 番目までの奇数の 2 乗の和を求める関数 sumsqodd を，if による場合分けを用いて，関数 sumsqodd2 として作成せよ.

解答例. 以下の通り.

```
sumsqodd2(n) = if n==1 then 1 else (2*n-1)^2 + sumsqodd2(n-1)
```

□

問題 27 (B 問題)**.** 実習資料 第 02 時限の問題 38 の x の n 乗 (x^n) を計算する関数 pow(x, n) を，if による場合分けを用いて，関数 pow2(x, n) として作成せよ.

解答例. 以下の通り.

```
pow2(x, n) = if n==0 then 1 else x * pow2(x, n-1)
```

□

問題 28 (B 問題). ニュートン法により $\sqrt{5}$ を計算する関数 root5(n) を定義せよ. ただし, n が 0 のときの初期値を 5 とする.

解答例. 以下の通り.

```
root5(0) = 5
root5(n) = root5(n-1)/2 + 5/(2*root5(n-1))
```

□

問題 29 (B 問題). ニュートン法により $\sqrt{x}$ を計算する関数 root(x, n) を定義せよ. ただし, n が 0 のときの初期値を x とする. 関数 root を用いて $\sqrt{599}$ を計算し, n が大きくなると, その値の 2 乗が 599 に近づくことを確認せよ.

解答例. 以下の通り.

```
root(x, 0) = x
root(x, n) = root(x, n-1)/2 + x/(2*root(x, n-1))
```

□

問題 30 (D 問題). 実際にニュートン法で近似解を求める際に, ある微小な値 ε を与えて, $|x_n - x_{n-1}| < \varepsilon$ となったときに, x_n を計算に用いるのに十分な近似解とすることがある.

問題 29 の関数 root(x, n) を使い, 微小な値 eps で, $\sqrt{x}$ を求める関数 root_eps(x, eps) を, 補助関数 root_eps_n(x, eps, n) を用いて次のように定義できる.

```
{- n は 1 以上の整数. 関数 myabs は実習資料~第 02 時限の絶対値を返す関数 -}
root_eps_n(x, eps, n) = if myabs(root(x,n) - root(x,n-1)) < eps
                        then root(x, n)
                        else root_eps_n(x, eps, n+1)

root_eps(x, eps) = root_eps_n(x, eps, 1)
```

例えば, $\varepsilon = 0.000001$ で, $\sqrt{13}$ を求めるには, root(13, 0.000001) とすればよい.

関数 root_eps_n は本体で root_eps_n 自身を呼び出しているので, 再帰関数である. これまでの実習で扱った再帰関数と root_eps_n の違いについて考察せよ.

解答例．これまでの再帰関数は n>=1 の引数が与えられたとき，引数の n を減らしながら再帰呼出しをし，n が特定の値になったとき (例えば，n==0 や n==1 のとき)，直接，値を返して再帰呼出しが停止する．root_eps_n は引数 n を増やしながら再帰呼出しをし，n が特定の値になったときではなく，ある条件 (myabs(root(x,n)-root(x,n-1)) が eps 未満) を満たしたときに，再帰呼出しが停止する． □

問題 31 (D 問題)**．** 与えられた 2 つの正整数の最大公約数を求める関数 mygcd を作成せよ．なお，2 つの正整数 $m, n (m \geq n)$ について，次のようにして最大公約数を求めることができる (ユークリッドの互除法)．

- $n = 0$ ならば，m が求める最大公約数である．
- そうでなければ，求める最大公約数は，n と，m を n で割った余りの最大公約数である．

解答例．以下の通り．

```
mygcd(m,0) = m
mygcd(m,n) = mygcd(n, m `mod` n)
```

□

問題 32 (A 問題)**．** 値 n が偶数であれば True を，そうでなければ False を返す関数 isEven を定義せよ．なお，True, False は Haskell で真偽を表す値 (真理値，論理値) である．

解答例．以下の通り．

```
isEven(n) = if n `mod` 2 == 0 then True else False
```

□

問題 33 (B 問題)**．** 値 n が奇数であれば True を，そうでなければ False を返す関数 isOdd を定義せよ．

解答例．以下の通り．

```
isOdd(n) = if n `mod` 2 == 1 then True else False
```

別解: `isOdd(n) = if n `mod` 2 == 0 then False else True`

□

問題 34 (B 問題)**．** 値 x が集合の要素であれば True を，そうでなければ False を返す関数 memberOf を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$ に対して 3 および 6 を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
memberOf(x, [])   = {- 空集合の場合 -}
memberOf(x, a:as) = {- 空集合でない場合 -}
```

解答例. 以下の通り．

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

memberOf(3, [1,2,3,4,5]) = True
memberOf(6, [1,2,3,4,5]) = False
```

□

問題 35 (B 問題)**.** 集合の要素数を求める関数 numberOf を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$，$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$，および空集合を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
numberOf([])   = {- 空集合の場合 -}
numberOf(a:as) = {- 空集合でない場合 -}
```

解答例. 以下の通り．

```
numberOf([])   = 0
numberOf(a:as) = 1 + numberOf(as)

numberOf([1,2,3,4,5]) = 5
numberOf([1,2,3,4,5,6,7,8,9,10]) = 10
numberOf([]) = 0
```

□

問題 36 (B 問題)**.** 集合に含まれる要素の総和を求める関数 sumSet を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$，$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$，お

よび空集合を評価した結果を確認せよ.

イコール以降を記述し，完成させなさい.

```
sumSet([])   = {- 空集合の場合 -}
sumSet(a:as) = {- 空集合でない場合 -}
```

解答例. 以下の通り.

```
sumSet([])   = 0
sumSet(a:as) = a + sumSet(as)

sumSet([1,2,3,4,5]) = 15
sumSet([1,2,3,4,5,6,7,8,9,10]) = 55
sumSet([]) = 0
```

□

問題 37 (B 問題). 集合に含まれる要素の最大値を求める関数 maxSet を定義せよ. 下記のように，集合の要素が 1 つだけの場合と，そうでない場合に分けて定義すること. また，集合 $\{3, 1, 5, 2, 9, 7, 4, 10, 6, 8\}$ を評価した結果を確認せよ.

イコール以降を記述し，完成させなさい.

```
maxSet([a])  = {- 要素が 1 つだけの場合 -}
maxSet(a:as) = {- 複数の要素がある場合 -}
```

解答例. 以下の通り.

```
maxSet([a]) = a
maxSet(a:as) = if a > maxSet(as) then a
               else maxSet(as)

maxSet([3,1,5,2,9,7,4,10,6,8]) = 10
```

□

問題 38 (B 問題). 集合に含まれる要素の平均を求める関数 averageSet を定義せよ. これまでの問題で作成した関数を使用してよい. また，集合 $\{1, 2, 3, 4, 5\}$ と $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ を評価した結果を確認せよ.

解答例．以下の通り．

```
averageSet(as) = sumSet(as) / numberOf(as)

averageSet([1,2,3,4,5]) = 3.0
averageSet([1,2,3,4,5,6,7,8,9,10]) = 5.5
```

□

問題 39 (B 問題)．集合に含まれる要素の最小値を求める関数 minSet を定義せよ．下記のように，集合の要素が 1 つだけの場合と，そうでない場合に分けて定義すること．また，集合 $\{3, 1, 5, 2, 9, 7, 4, 10, 6, 8\}$ を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
minSet([a])  = {- 要素が 1 つだけの場合 -}
minSet(a:as) = {- 複数の要素がある場合 -}
```

解答例．以下の通り．

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

minSet([3,1,5,2,9,7,4,10,6,8]) = 1
```

□

問題 40 (B 問題)．集合に含まれる要素のうち，正の値の個数を求める関数 countPositives を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{-3, -2, -1, 0, 1, 2, 3\}$ を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
countPositives([])   = {- 空集合の場合 -}
countPositives(a:as) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countPositives([]) = 0
countPositives(a:as) = if a > 0
                       then 1 + countPositives(as)
                       else countPositives(as)

countPositives([-3,-2,-1,0,1,2,3]) = 3
```

□

問題 41 (B 問題)．集合に含まれる要素のうち，偶数の個数を求める関数 countEven を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$ を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
countEven([])   = {- 空集合の場合 -}
countEven(a:as) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countEven([]) = 0
countEven(a:as) = if a `mod` 2 == 0
                  then 1 + countEven(as)
                  else countEven(as)

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

□

問題 42 (B 問題)．集合に含まれる要素のうち，奇数の個数を求める関数 countOdd を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$ を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
countOdd([])   = {- 空集合の場合 -}
countOdd(a:as) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countOdd([]) = 0
countOdd(a:as) = if a ‘mod‘ 2 == 1
                   then 1 + countOdd(as)
                   else countOdd(as)

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

□

問題 43 (B 問題)**.** 集合に含まれる要素のうち，3 の倍数の個数を求める関数 `countMultiplesOf3` を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ を評価した結果を確認せよ．

イコール以降を記述し，完成させなさい．

```
countMultiplesOf3([])   = {- 空集合の場合 -}
countMultiplesOf3(a:as) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countMultiplesOf3([]) = 0
countMultiplesOf3(a:as) = if a ‘mod‘ 3 == 0
                   then 1 + countMultiplesOf3(as)
                   else countMultiplesOf3(as)

countMultiplesOf3([1,2,3,4,5,6,7,8,9,10]) = 3
```

□

問題 44 (B 問題)**.** 集合に含まれる要素のうち，値 x 以上の個数を求める関数 `countEqualAndGreaterThan` を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ および，$\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ に対して x を 60 として評価した結果をそれぞれ確認せよ．

イコール以降を記述し，完成させなさい．

```
countEqualAndGreaterThan([], x)   = {- 空集合の場合 -}
countEqualAndGreaterThan(a:as, x) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countEqualAndGreaterThan([], x) = 0
countEqualAndGreaterThan(a:as, x) = if a >= x
                                    then 1 + countEqualAndGreaterThan(as, x)
                                    else countEqualAndGreaterThan(as, x)

countEqualAndGreaterThan([10,20,30,40,50,60,70,80,90,100],60) = 5
countEqualAndGreaterThan([75,54,98,72,45,60,88,59,92,55,35],60) = 6
```

□

問題 45 (B 問題)**.** 値 x が 2 の倍数であり，かつ 3 の倍数である場合 `True` を，そうでなければ `False` を返す関数 `isMultipleOf2And3` を定義せよ．また，x を 1 から 6 まで順に評価した結果をそれぞれ確認せよ．

Haskell では 2 つの条件式を`&&`で組み合わせることで，すべての条件を満たすかどうかを判定することができる．また，`||`で組み合わせれば，すべての条件のうち少なくとも一つを満たす場合を判定できる．例えば，$0 \leq x < 10$ という条件は `0 <= x && x < 10` と記述し (`0 <= x < 10` は誤り)，$x < 0$ または $10 \leq x$ という条件は `x < 0 || 10 <= x` と記述する．

解答例．以下の通り．

```
isMultipleOf2And3(x) = if x `mod` 2 == 0 && x `mod` 3 == 0
                       then True else False
isMultipleOf2And3(1) = False
isMultipleOf2And3(2) = False
isMultipleOf2And3(3) = False
isMultipleOf2And3(4) = False
isMultipleOf2And3(5) = False
isMultipleOf2And3(6) = True
```

□

問題 46 (B 問題)**.** 値 a が x 以上 y 以下の範囲内であれば `True` を，そうでなければ `False` を返す関数 `isRange(a, x, y)` を定義せよ．また，x を 2，y を 6 とした上で，a を 1，3，および 7 と変えて評価した結果をそれぞれ確認せよ．

解答例．以下の通り．

```
isRange(a, x, y) = if a >= x && a <= y
                   then True else False

isRange(1,2,6) = False
isRange(3,2,6) = True
isRange(7,2,6) = False
```

□

問題 47 (B 問題)．値 x が 2 または 3 の倍数であれば `True` を，そうでなければ `False` を返す関数 `isMultipleOf2Or3` を定義せよ．また，x を 1 から 6 まで順に評価した結果をそれぞれ確認せよ．

解答例．以下の通り．

```
isMultipleOf2Or3(x) = if x `mod` 2 == 0 || x `mod` 3 == 0
                      then True else False

isMultipleOf2Or3(1) = False
isMultipleOf2Or3(2) = True
isMultipleOf2Or3(3) = True
isMultipleOf2Or3(4) = True
isMultipleOf2Or3(5) = False
isMultipleOf2Or3(6) = True
```

□

問題 48 (B 問題)．集合に含まれる要素のうち，x 以上 y 以下の範囲内の個数を求める関数 `countRange` を定義せよ．下記のように，空集合の場合と，そうでない場合に分けて定義すること．また，集合 $\{1, 2, 3, 4, 5\}$ に対して x を 2，y を 4 として評価した結果と，集合 $\{75, 54, 98, 72, 45, 60, 88, 59, 92, 55, 35\}$ に対して x を 60，y を 80 として評価した結果をそれぞれ確認せよ．

イコール以降を記述し，完成させなさい。

```
countRange([], x, y)   = {- 空集合の場合 -}
countRange(a:as, x, y) = {- 空集合でない場合 -}
```

解答例．以下の通り．

```
countRange([], x, y) = 0
countRange(a:as, x, y) = if a >= x && a <= y
                         then 1 + countRange(as, x, y)
                         else countRange(as, x, y)

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

□

問題 49 (B 問題)．100 点満点の点数 (Score) の集合において，問題 17 で作成した関数 s2gp で評価した結果 (Grade Point) が 0 以上 (点数が 0 未満や 100 より大でない) となる要素について，その評価後の結果 (Grade Point) の合計値を計算する関数 gpt を定義せよ．また，集合 {75, 54, 98, 104, 72, 45, -13, 60, 88, 59, 92, 55, 35} に対して評価した結果を確認せよ．

解答例．以下の通り．

```
gpt([]) = 0
gpt(a:as) = if s2gp(a)>=0 then s2gp(a) + gpt(as) else gpt(as)

gpt([75,54,98,104,72,45,-13,60,88,59,92,55,35]) = 16
```

□

問題 50 (B 問題)．100 点満点の点数 (Score) の集合において，問題 17 で作成した関数 s2gp で評価した結果 (Grade Point) が 0 以上 (点数が 0 未満や 100 より大でない) となる要素の個数を計算する関数 gpnum を定義せよ．また，集合 {75, 54, 98, 104, 72, 45, -13, 60, 88, 59, 92, 55, 35} に対して評価した結果を確認せよ．

解答例．以下の通り．

```
gpnum([]) = 0
gpnum(a:as) = if s2gp(a)>=0 then 1 + gpnum(as) else gpnum(as)

gpnum([75,54,98,104,72,45,-13,60,88,59,92,55,35]) = 11
```

□

問題 51 (B 問題)．100 点満点の点数 (Score) の集合において，問題 17 で作成した関数 s2gp で評価した結果 (Grade Point) が 0 以上 (点数が 0 未満や 100 より大でない) となるの要素の平均値を計算する関数 gpa を定

義せよ．ただし，0 以上の結果がないことは想定しなくて良い．また，集合 {75, 54, 98, 104, 72, 45, -13, 60, 88, 59, 92, 55, 35} に対して評価した結果を確認せよ．

解答例．以下の通り．

```
gpa(as) = gpt(as)/gpnum(as)
```

```
gpa([75,54,98,104,72,45,-13,60,88,59,92,55,35]) = 1.4545454545454546
```

□