

プログラミング・パラダイム（2018年度）・テスト問題用紙

(2018年11月29日(木)・9:35～10:20)

解答上、その他の注意事項

- I. 問題は、問 I～II までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. ノート・プリント・参考書などは持ち込み可である。
- IV. 携帯電話などの通信機能を持つものは 持ち込み不可 である。
- V. 問 I を解答するときのみ、PC を使用して良い。ネットワークに接続して WWW を閲覧しても良いが、掲示板、チャット、メールなどで生身の人間と通信することは禁じる。
- VI. テストの配点は 50 点である。合格はレポートの得点を加えて、100 点満点中 60 点以上とする。

- (1) 引数として与えられる整数の組のリストの要素のなかで、すべての組の第 2 要素が第 1 要素を割り切るかどうかを返す関数

```
foo :: [(Integer,Integer)] -> Bool
```

を定義せよ。

例えば、foo [(12,5),(9,3)]は False であり、foo [(2,1),(10,5),(9,3)]は True である。

この問では map, filter, foldl, foldr などのリストに関するライブラリ関数や内包表記を使わず、if ~ then ~ else ~ 式、算術演算子、論理演算子、比較演算子、パターンマッチング、ガード、再帰のみを使って定義せよ。

また、Haskell で“余り”を表す算術演算子は ‘mod’ である。例えば 7 ‘mod’ 4 (あるいは mod 7 4) の値は 3 である。

- (2) 正の整数 n を引数として受け正の整数の組 (i, j) で、 $1 \leq i \leq n$ 、 $1 \leq j \leq n$ で $2i + j \leq n$ かつ $i + 2j \leq n$ となるものを列挙する関数

```
bar :: Integer -> [(Integer,Integer)]
```

を (リストの内包表記を用いて) 定義せよ。

例えば、bar 2 は []、bar 4 は [(1,1)] で、bar 6 は [(1,1),(1,2),(2,1),(2,2)] となる。

(リストの要素の順番はこの通りでなくても良い。)

なお、 m から n まで (ただし $m \leq n$) の整数のリストは、[m..n] という式で得ることができる。例えば、[0..5] は [0,1,2,3,4,5] である。

問 I の解答が完了した人は PC を閉じるかシャットダウンし、挙手して問 II の問題用紙を受け取ること。

II. (Haskell)

(12点×2)

次の例にならって、下の Haskell の式 (1)~(2) を評価した結果を書け。

例: `take 5 (from 1)` ⇒ 評価結果: `[1,2,3,4,5]`

ただし、`take` と `from` は講義プリントに定義されている通りの関数である。

```
from :: Integer -> [Integer]
from n = n : from (n + 1)

take :: Integer -> [a] -> [a]
take 0 _      = []
take _ []     = []
take n (x:xs) = x : take (n - 1) xs
```

(1) `takeWhile (< 90) (map (\ x -> 2 * x * x) (from 1))`

この問で使用されている関数 `map`, `takeWhile` の定義は次の通りである。

```
map :: (a -> b) -> [a] -> [b]
map f []      = []
map f (x:xs) = f x : map f xs

takeWhile :: (a -> Bool) -> [a] -> [a]
takeWhile p []      = []
takeWhile p (x:xs) = if p x then x : takeWhile p xs
                    else []
```

(2) `[(x,y) | x <- [3..6], y <- [2..4], x 'mod' y == 0]`

(この問に関してはリスト内の順番のみの間違いは、減点はしない。)

