

プログラミング・パラダイム（2020年度）・テスト問題用紙
（2020年11月26日（木）・08:50～10:20）

解答上、その他の注意事項

1. 問題は、問 I～II までである。
2. 解答用紙の右上の欄に学籍番号・名前を記入すること。
3. ノート・プリント・参考書などは持ち込み可である。
4. 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。
5. 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。
6. テストの配点は50点である。合格はレポートの得点を加点して、100点満点中60点以上とする。

I. (Haskell 実習問題)

1. 引数として与えられる整数のペアのリストの要素のなかで、第 1 成分が偶数であるものの、第 2 成分の和を返す関数

```
1 foo :: [(Integer,Integer)] -> Integer
2
```

を定義せよ。

例えば、foo [(12,5),(9,9),(2,-3)] は 2 であり、foo [(1,5),(2,4),(4,-1),(8,3),(3,2)] は 6 であり、foo [(3,1),(5,10),(7,3)] は 0 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 $7 \bmod 4$ (あるいは $\text{mod } 7 \ 4$) の値は 3 である。

2. 正の整数 n を引数として受け、 $0 \leq i \leq j \leq n$ を満たす整数の組 (i, j) で、 $i^2 - j$ が 3 の倍数であるものを列挙する関数

```
1 bar :: Integer -> [(Integer,Integer)]
2
```

を (リストの内包表記を用いて) 定義せよ。

例えば、bar 2 は [(0,0),(1,1)]、bar 4 は [(0,0),(0,3),(1,1),(1,4),(2,4),(3,3),(4,4)] で、bar 6 は [(0,0),(0,3),(0,6),(1,1),(1,4),(2,4),(3,3),(3,6),(4,4),(6,6)] 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。

また、Haskell で“余り”を表す算術演算子は `mod` である。例えば $7 \bmod 4$ (あるいは $\text{mod } 7 \ 4$) の値は 3 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、問 II の問題用紙を受け取ること。

II. (Haskell)

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。

```
1 from :: Integer -> [Integer]
2 from n = n : from (n + 1)
3
4 take :: Integer -> [a] -> [a]
5 take 0 _ = []
6 take _ [] = []
7 take n (x:xs) = x : take (n - 1) xs
8
```

1. `foldr (\ x y -> x + 3 * y) 0 [1,3,2]`

この問で使用されている関数 `foldr` の定義は次の通りである。

```
1 foldr :: (a -> b -> b) -> b -> [a] -> b
2 foldr f z [] = z
3 foldr f z (x:xs) = f x (foldr f z xs)
4
```

2. `[(x,y) | x <- [1..6], y <- [1..x], 6 * y >= x * x]`

(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。)

