

プログラミング言語論 中間テスト (2023 年度)
テスト問題用紙
(2023 年 11 月 30 日 (木) ・ 10:30 ~ 12:00)

解答上、その他の注意事項

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

II. (Haskell)

次の例にならって、下の Haskell の式 1.~ 4. を評価した結果を書け。

例: `take 5 [1..]` ⇒ **評価の結果:** `[1,2,3,4,5]`

ただし、`take` は講義プリントに定義されている通りの関数である。

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

1. `take 6 (filter (\ x -> (x * x) `mod` 3 == 1) [1..])`

この問で使用されている関数 `filter` の定義は次の通りである。また `x `mod` y` は `x` を `y` で割った余りである。

```
1 filter :: (a -> Bool) -> [a] -> [a]
2 filter _ []         = []
3 filter p (x:xs)    = if p x then x : filter p xs else filter p xs
```

2. `take 6 (iterate (\ (x, y) -> (y, 2 * x - y)) (2, 1))`

この問で使用されている関数 `iterate` の定義は次の通りである。

```
1 iterate :: (a -> a) -> a -> [a]
2 iterate f a = a : iterate f (f a)
```

3. `let foo = zipWith (+) [1..] (1 : 2 : foo) in take 6 foo`

この問で使用されている関数 `zipWith` の定義は次の通りである。

```
1 zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]
2 zipWith f (x:xs) (y:ys) = f x y : zipWith f xs ys
3 zipWith _ _ _          = []
```

4. `[(x,y) | x <- [1..5], y <- [x..(x+3)], y >= 2 * x - 1]`

(この問に関してはリスト内の順番の間違いは、減点はしない。)

