プログラミング言語**特**論・テスト問題用紙

('06 年 2 月 9 日（木）・ 13:00 〜 14:30)

解答上、その他の注意事項

I. 問題は、問 I〜IV まである。

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. テストの配点は 50 点（+ボーナス 20 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. （ラムダ計算） （7 点 × 2）

次の λ 式が正規形に到達するまでの、最左変換による 1 ステップずつの β 変換の列を書け。ただし、正規形が存在しない式については、それが判別できる時点（ただし少なくとも 3 回以上 β 変換したあと）に $\cdots$ と記入せよ。

記入例:

$$
\begin{array}{rl}
 & (\lambda f x.f(fx))((\lambda f x.f(fx))g)y \\
\xrightarrow{\beta} & (\lambda x.((\lambda f x.f(fx))g)(((\lambda f x.f(fx))g)x))y \\
\xrightarrow{\beta} & ((\lambda f x.f(fx))g)(((\lambda f x.f(fx))g)y) \\
\xrightarrow{\beta} & (\lambda x.g(gx))(((\lambda f x.f(fx))g)y) \\
\xrightarrow{\beta} & g(g(((\lambda f x.f(fx))g)y)) \\
\xrightarrow{\beta} & g(g((\lambda x.g(gx))y)) \\
\xrightarrow{\beta} & g(g(g(gy)))
\end{array}
\qquad
\begin{array}{rl}
 & (\lambda x.xx)(\lambda x.xx) \\
\xrightarrow{\beta} & (\lambda x.xx)(\lambda x.xx) \\
\xrightarrow{\beta} & (\lambda x.xx)(\lambda x.xx) \\
\xrightarrow{\beta} & (\lambda x.xx)(\lambda x.xx) \\
\xrightarrow{\beta} & \cdots
\end{array}
$$

(1) $(\lambda x.x(\lambda xy.x))(\lambda x.x(\lambda xy.y))(\lambda x.x)$

(2) $(\lambda y.y((\lambda a.xa)(\lambda a.a)))(\lambda b.b)$

なお、必要に応じて $I = \lambda x.x$ など適宜、定数を定義しても良い。

II. （Haskell） （6 点 × 2）

次の Haskell のプログラムを評価した結果を書け。

例: `take 5 (from 1)` ⇒ **答:** `[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とおりの関数である。

```
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) `take 5 (foo 0 (iterate (\ x -> 2 * x) 1))`

(2) `[ x*y | x <- [1, 2, 3], y <- [4, 5, 6], even (x+y)]`
（この問に関してはリスト内の順番の間違いは -1 点とする。）

ただし、この問で使用されている関数の定義は次のとおりである。

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

foo :: Integer -> [Integer] -> [Integer];
foo i []     = [];
foo i (x:xs) = (i+x) : foo (i+x) xs;
```

また、`even` は偶数かどうかを判定する述語である。

III. （語句） （8 点 × 3）

プログラミング言語（やその処理系）で用いられる次の 6 つの語句のうち 3 つを選択し、具体的な例を挙げて説明せよ。ただし、講義プリントにのっている例ではなく、オリジナル の例を考えること。

- 動的束縛（dynamic binding）
- 遅延評価（lazy evaluation）
- カプセル化（encapsulation）
- CPS（continuation passing style）
- 多相（polymorphism）
- 多重定義（overloading）

IV. （自由記述 — ボーナス問題） （最高 20 点）

これまでにいろいろなプログラミング言語を学習して、理解しにくかった点、間違いやすいと感じた点、あるいは初心者がつまづきやすいと感じた点は何か? できるだけ具体的に述べよ。また、それに対して、オリジナルの改良のアイデアや抜本的な解決策を（実現性は気にせず）自由に考述せよ。

プログラミング言語**特**論・テスト解答用紙('06年2月9日)

学籍番号		氏名	

学籍番号		氏名	