

## # 2020年度非手続き型言語過去問 オンラインによるテストでした

関数型言語と手続き型言語との違いとして間違っているものはどれか。

1つ選択してください:

- 関数型言語では関数の再帰呼び出しができる。
- 関数型言語にはelse文のないif文は存在しない。
- 関数型言語ではプログラムの途中で変数の値を変えることはできない。
- 関数型言語にはループ文がない。

組  $(4.0, [3])$  は次のどのパターンとマッチするか？1つまたはそれ以上選択してください:

- $(n, [x])$
- $(i, l)$
- $(i, x::nil)$
- $(i, x::y::xs)$
- $(n, x::xs)$

Warning: match nonexhaustive とはどういう意味か？1つ選択してください:

- パターンの書き方が間違っているという警告
- 式が書かれていないパターンがあるという警告
- パターンの書き方が間違っているというエラー
- 式が書かれていないパターンがあるというエラー

第6回講義の課題にあったよう二変数の多項式をリストのリストで表現する。次の式の表現として正しいものはどれか。

$$x^2 + y^2 + 3xy + x + 5y + 4$$

1つ選択してください:

- a.  $[[4], [1, 5], [3], [1, 1]]$
- b.  $[[4, 5, 1], [1, 3], [1]]$
- c.  $[[4.0], [1.0, 5.0], [3.0], [1.0, 1.0]]$
- d.  $[[4.0, 5.0, 1.0], [1.0, 3.0], [1.0]]$

以下の中でエラーとならず正しく関数が定義できるものはどれか。1つまたはそれ以上選択してください:

```
fun f (nil) = nil
| f (x::xs) = (hd (x))::f(xs);
```

```
fun f (nil) = nil
| f (x::xs) = (abs x)::f(xs);
```

```

fun f (nil) = nil
| f (x::xs) = (f (x))::f(xs);

fun f (nil) = nil
| f (x::xs) = (explode (x))::f(xs)

```

次の中でリストを逆順にする関数として正しくないものはどれか。1つまたはそれ以上選択してください:

```

fun rev1 (M, []) = M
| rev1 (M, x::xs) = rev1(x::M, xs);
fun rev (L) = rev1 (L, []);

fun rev1 (M, []) = M
| rev1 (M, x::xs) = rev1(x::M, xs);
fun rev (L) = rev1 ([], L);

fun reverse ([]) = []
| reverse (x::xs) = xs@[x];

fun reverse ([]) = []
| reverse (x::xs) = reverse(xs)@[x];

```

次の中で末尾再帰になっていないものはどれか。1つ選択してください:

```

fun f (x::nil) = x
| f (x::xs) = f (xs);

fun f(n) =
let
fun f1 (s,1) = s
| f1 (s,i) = f1 (is, i-1)
in
f1(1,n)
end;

fun f (x, i)=
let
fun f1(s,0) = s
| f1 (s,i) = f1 (sx, i-1)
in
f1(1.0, i)
end;

fun f (n) =
let
fun f1 (s, 0) = 0
| f1 (s, i) = s + f1(s-1, i-1)

```

```
in
f1 (n, n)
end;
```

次の関数は何をする関数か。結果は対になるが、その第一要素、第二要素が何であるかも併せて答えよ。

```
fun f ([]) = ([], [])
| f ([a]) = ([a], [])
| f (a::b::cs) =
  let
    val (M, N) = f(cs)
  in
    (b::M, a::N)
  end;
```

次の関数は型エラーが起こって定義できない。型エラーを起こさないように修正したものを答えよ。また修正した関数は何を行う関数かを答えよ。

```
fun f (a, nil) = nil
| f (a, x::nil) = a::x::nil
| f (a, x::xs) = (a::x)::xs;
```

実数のリストの最大値（つまり、要素の値のうちで最大のもの）を返す関数を書け。リストの長さに比例した時間で計算できるものである方が望ましい。

## 後

---

台形則を使って積分を計算する関数trapを授業内では以下のように定義した。

```
fun trap (a, b, n, F) =

  if n<=0 orelse b-a <= 0.0 then 0.0

  else

    let

      val delta = (b-a) / real (n)

    in

      delta * (F(a) + F(a+delta))/2.0 + trap (a+delta, b, n-1, F)
```

```
end;
```

これを使って例えば

```
trap(0.0, 1.0, 10, (fn (x:real) => x * x));
```

とすれば積分値を計算することができる.

関数trapをカリー化形式で書き直すと,  $n$  の値を引数として, ある決まった範囲の関数 $F$ の積分値を計算する関数 $T$ を定義することができる. そのために関数trapをどのようなカリー化形式の関数にすればよいかを書け.

(つまり,  $\text{real} \rightarrow \text{real} \rightarrow (\text{real} \rightarrow \text{real}) \rightarrow \text{int} \rightarrow \text{real}$  となるカリー化形式の関数trapを書けということ)

以下は関数 $s$ の型の推定を行なっている. 穴を埋めて推定を完成せよ.

```
fun S x y z = (x z) (y z)
```

次のように型変数を割り当てる.

(いつもは $t$ を使っているが, 今回は入力しやすいように $t_1, t_2, \dots$ を用いる)

$S: t_1, x: t_2, y: t_3, z: t_4, (x z): t_5, (y z): t_6, (x z) (y z): t_7.$

求めたいのは  $t_1 = t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow t_7.$

関数 $x$ について  $t_2 = \text{回答} \rightarrow \text{回答}$

関数 $y$ について  $t_3 = t_4 \rightarrow t_6$

関数 $(x z)$ について  $t_5 = \text{回答} \rightarrow \text{回答}$

これらを代入すると

```
t1 = (回答 -> t5) -> (t4 -> t6) -> t4 -> t7
    = (t4 -> t6 -> 回答) -> (t4 -> t6) -> t4 -> t7
```

よって

```
('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c
```

リストの奇数番目と偶数番目の要素を入れ換える関数を書け. すなわち, リスト $[a_1, a_2, \dots, a_n]$ が与えられると $[a_2, a_1, a_4, a_3, a_6, a_5, \dots]$ を生じる. もし $n$  が奇数ならば $a_n$  がそのまま最後の要素となる.

- map関数で実現できるのはどれか？
  - 1. リストの小文字を大文字にするbigletter
  - 2. リストの要素の各和
  - 3.  $x$ の $i$ 乗
  - 4. (すまない, 覚えてない・・・)
- 
- 上記の4つのうち, 1つを実現する関数をかけ.