

人工知能プログラミング 2015 年問題(再現)

※問題は回収。再現のため例とか問題文に若干違う部分あり

配点は問 5, 問 6 のみ各 20 点、それ以外各大問 10 点

解答の際に授業中にプリント等で定義した関数を使っても良い

問1. 2 引数のリスト内における全要素から 2 引数のリストに共通する要素を省く **exor** 関数を作れ 例: (**exor** '(a b c d) '(b d f h)) → (A C F H)

問2. 任意の深さを持つリストを引数にとり、リスト内の数値を絶対値に変換する **allabs** 関数を作れ。ただしリストの構造は変えてはいけないものとし、数値以外の要素についてはそのまま出力せよ。例: (**allabs** '((-1) 200 ((abc))))) → ((1) 200 ((ABC)))

問3. 関数 **func** とリスト **x** を引数にとり、**x** の要素を 1 つずつ **func** に適用し最初に真となった要素を返す **myfindif** 関数を作れ。ただし以下のコードの続きを書け。

例: (**myfindif** #oddp '(1 2 3))→1 (**myfindif** #oddp '(2 4 6))→NIL

(**myfindif** #(lambda (x) (> x 3)) '(1 2 3 4 5 6))→4

<コード>

```
(defun myfindif (func x)
  (cond
```

問4. 授業中に作った **drawline** 関数を用いて引数の数値の行数の大きさの三角形を出力する **triangle** 関数を作れ。また、作ったコードによりどのような三角形ができるか概略図を示すこと。

例:(**triangle** 5)→\*

	*****	*	*
**	****	**	**
***	***	***	***
****	**	****	****
*****	*	*****	*****

のどれか

問5. 要素 2 つのリストで(強さ マーク)を表すとする。

強さは 2,3,4,5,6,7,8,9,10,jack,queen,king,ace の 13 種類、

マークは club,spade,heart,diamond の 4 種類あるとすることでこのリストでランプを表現することができる。

- (1) このリストを引数に取り、そのカードの強さを出力する **rank** 関数、そのカードのマークを出力する **suit** 関数をつくれ。

例: (rank '(ace heart))→ACE (suit '(ace heart))→HEART

- (2) 以下のように手札のリストを大域変数で定義したとする。

(setf my-hand '((4 club) (3 heart) (jack club)))

マークと手札のリストを引数にとり、手札内のそのマークのカードの枚数を数える **count-suit** 関数を作れ

例: (count-suit club)→2 (count-suit diamond)→0

- (3) **remove-if-not** を使い、手札のリストを引数に取り、手札内の黒色のカードのみをリストとして表示する **black-cards** 関数を作れ

例: (black-cards my-hand)→((4 CLUB) (JACK CLUB))

問6. 授業で扱ったブロックの世界に関する問題である(当日の問題にはブロックの世界についての **blocklist.lsp** の入手方法も書かれている。)

- (1) 引数を 1 つ取り、その引数の形をしたブロックを取り出す関数 **shapeblock** を作れ 例: (shapeblock 'brick)→(B1 B2 B3 B6) (shapeblock 'ball)→NIL

- (2) 引数を 1 つ取り、その引数のブロックと同じ色のブロックをリストとして返す **samecolor** 関数を作れ

例: (samecolor 'b1)→(B5) ※自身は結果から除くようにすること

(samecolor 'b4)→NIL

問7. 授業で扱った三目並べに関する問題である。授業で扱ったコードではコンピュータは 3 つの戦略を使うようにプログラムされていたが、これに新たに真ん中が空いていればそこを打つという戦略を加えたい。

(defun choose-best-move (board)

(or (make-three-in-a-row board) ; 第一の戦略

(block-human-win board) ; 第二の戦略

(put-center board) ; 新しい戦略！！

(random-move board))) ; 最後の手段

新しい戦略となる関数 **put-center** 関数を作れ。(この問題も三目並べのプログラムの入手方法は書かれていた)

問8. `append` 関数について以下のような操作を行った。

`(setf x a)`

`(setf y '(a))`

`(setf z '(b c))`

`(append y z)→(A B C)`

`(append x z)→error: x is not a list.`

`(append z x)→(B C . A)`

`(eq y (append y z))→NIL`

`(eq z (rest (append y z)))→T`

以上のような結果から `append` は 2 引数をどのようにつなげていると予想できるか  
答えよ。必要ならば図を用いて説明しても良い。