

## 1 動作トレース (その1)

```
struct List {                                | (設問)
    int      data ;                          | 以下のプログラムを動かした場合に
    struct List* next ;                      | (a) によって途中に出力される内容を答えよ。
};                                            | (b) によって出力される内容を答えよ。

struct List* uniq_list( struct List* p ) {   |
    struct List* q ;                         |
    for( q = p ; q != NULL ; q = q->next ) { |
        struct List* r = q->next ;          |
        struct List** s = &( q->next ) ;    |
                                                | (A)
        while( r != NULL ) {                |
            printf( "(%d:%d)", q->data, r->data ) ; |
            ~~~~~(a)
            if ( q->data == r->data ) {      |
                ~~~~~(B)
                struct List* d = r ;
                *s = r->next ;
                ~~~~~(C)
                r = r->next ;
                free( d ) ;
            } else {
                r = r->next ;
                s = &( (*s)->next ) ;
                ~~~~~(D)
            }
        }
    }
    return p ;
}

void main() {
    struct List* top
        = cons( 1 , cons( 2 , cons( 1 , NULL ) ) ) ;
    struct List* p = uniq_list( top ) ;
    printf( "\n" ) ;
    print_list( p ) ;
} ~~~~~(b)
```

```
struct List* cons( int x , struct List* n ) {
    struct List* ans =
        (struct List*)malloc(sizeof(struct List));
    if ( ans != NULL ) { ~~~~~(E)
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
} ~~~~~(F)

void print_list( struct List* p ) {
    for( ; p != NULL ; p = p->next )
        printf( "%d " , p->data ) ;
    printf( "\n" ) ; ~~~~~(G)
}
```

-----  
(回答欄)  
(a)  
  
(b)

## 2 動作トレース (その2)

前設問で、下線部 (A) ~ (E) の型が何か答えよ。

- (A) \_\_\_\_\_ (B) \_\_\_\_\_
- (C) \_\_\_\_\_ (D) \_\_\_\_\_
- (E) \_\_\_\_\_ (F) \_\_\_\_\_ ... (3x6)
- (G) \_\_\_\_\_ ... (2)

### 3 説明問題

以下に示すデータ構造について、データの出し入れの処理やそのイメージ図などを交えて説明せよ。

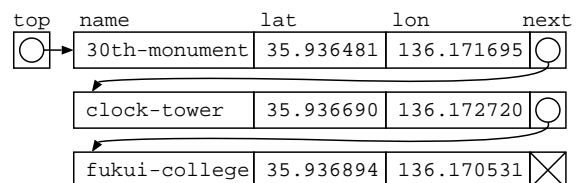
1. リストを用いた待ち行列
2. 配列を用いたスタック

### 4 プログラム記述基礎問題

場所の名前 (name), その緯度 (lat), 経度 (lon) の情報を図に示すようなリスト構造で、保存し以下のような処理を行いたい。(1) このデータに相応しいリスト構造の宣言を記述し、(2) このリストから、関数引数の緯度 (lat), 経度 (lon) と誤差が  $\pm 0.001$  度 (例 135.0 度なら、134.999-135.001, 緯度も同様とする) 以内の場所を出力する関数 `in_area()` を作成せよ。(20)

```
void main() {  
    緯度経度のリスト* top = .... ;  
    in_area( top , 35.936916 , 136.172854 ) ;  
    // "clock-tower 35.936690 136.172720"  
}
```

- (1) リストの宣言 (6)
- (2) `in_area` の記述 (14)

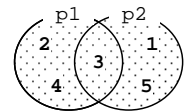
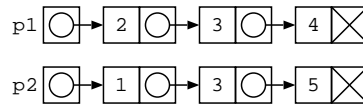


## 5 リスト基本説明

リスト構造と配列を比べ、リスト構造の利点を2つ、欠点を2つ、説明を交えて答えよ。(5x4)

## 6 リストで集合計算

リスト構造を用いて集合を表し、2つの集合の  
和集合を計算する関数を完成させよ。



```
list_union( (A) p1, (B) p2 ) {
    struct List* ans = p2 ;
    struct List* p ;
    struct List* q ;
    // 設問1の補助関数を使用あり。

    for( q = p1 ; (D) ; (E) ) {
        for( p = p2 ; (F) ; (G) ) {
            if ( (H) )
                break ;
            if ( (I) )
                ans = (J) ;
        }
    }
    return ans ;
}

void main() {
    struct List* p1 = cons(2, cons(3, cons(4, NULL))) ;
    struct List* p2 = cons(1, cons(3, cons(5, NULL))) ;
    struct List* p3 = list_union( p1 , p2 ) ;
}
```