

## 1 動作トレース (20)

以下のプログラムで表示される内容を答えよ。(5x4)

```
struct List {
    int data ;
    struct List* next ;
} ;

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

int find( struct List* p , int x ) {
    if ( p == NULL ) {
        return 0 ;
    } else {
        printf( "%d:%d\n" , p->data , x ) ;
        ~~~~~
        if ( p->data == x )
            return 1 ;
        else
            return find( p->next , x ) ;
    }
}

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

int uniq( struct List* p ) {
    struct List* pl ;
    for( pl = p ; pl != NULL ; pl = pl->next )
        if ( find( pl->next , pl->data ) )
            return 0 ;
    return 1 ;
}

void main() {
    struct List* top = cons(1,cons(2,cons(2,NULL))) ;
    printf( "%d\n" , uniq( top ) ) ;
    ~~~~~
}
```

## 2 リストによる集合処理 (20)

前述の設問の関数を使いながら、2つのリストによる集合で、集合積の計算を行うプログラムを作成せよ。

```
void main() {
    struct List* p1 = cons( 1 , cons( 2 , cons( 3 , NULL ) ) ) ;
    struct List* p2 = cons( 2 , cons( 4 , cons( 3 , NULL ) ) ) ;
    struct List* p3 = set_product( p1 , p2 ) ;
    print( p3 ) ; // 2,3 を表示 (表示順序は問わない)
}

struct List* set_product( struct List* p1 , struct List* p2 ) {
```

### 3 スタックと待ち行列プログラム

以下のプログラムにて、配列のスタック、リストによる待ち行列プログラムを作成した。

1. (A) スタック, (B) 待ち行列にふさわしい、英字 4 文字で表現せよ。(2)
2. プログラム中の (C), (G), (H) にふさわしい処理を答えよ。(3x3)
3. プログラム中の (D), (E), (F) の型を答えよ。(3x3)

```
int stack[ 10 ] ; /* (A) */      | struct List*top = NULL ; /* (B) */
int sp = 0 ;                    | struct List**tail = &top ;
void push( int x ) {           | void put( int x ) {
                                |     *tail = cons( x , NULL ) ;
                                |     tail = &( (*tail)->next ) ;
} ~~~~~(C)3                    | }
int pop() {                    | int get() {
    return stack[ --sp ] ;     |     struct List* d = top ;
} ~~~~~(D)                    |
void main() {                 |     int ans = ~~~~~(G)3
    push(1) ;                 |     top = ~~~~~(H)3
    push(2) ;                 |     free( d ) ;
    push(3) ;                 |     return ans ;
    printf( "%d\n" , pop() ) ; // 3 | }
    printf( "%d\n" , pop() ) ; // 2 | ((名称の回答欄)) (1x2)
                                | (A) _____ , (B) _____
    put(4) ; put(5) ; put(6) ;
    printf( "%d\n" , get() ) ; // 4
    printf( "%d\n" , get() ) ; // 5
}                               |
```

((型の回答欄)) (3x3)

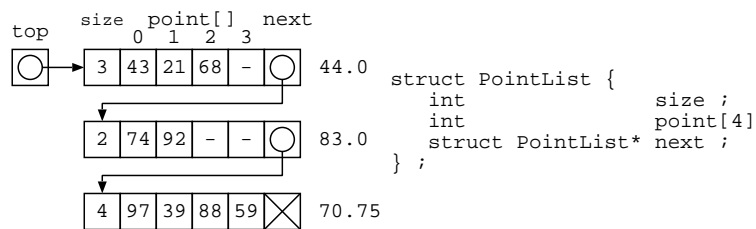
(D) \_\_\_\_\_ , (E) \_\_\_\_\_ , (F) \_\_\_\_\_

### 4 説明問題

1. リスト構造を用いたスタックへのデータの追加 (push) や取り出し (pop) について、図などを交えながら説明せよ。
2. 配列を用いた待ち行列へのデータの追加 (put) や取り出し (get) について、図などを交えながら説明せよ。

## 5 プログラム記述基礎問題 (20)

図に示すように、最大 4 科目の点数 (科目数は各データ毎に size に保存) データを、複数件、リスト構造で記録されている。各点数は 0~100 点とする。このリスト構造で、それぞれの平均点 (小数点あり) を求め、最高の平均点を出力する関数を作成せよ。



```
void print_max_mean( struct PointList* top ) {
```

## 6 説明問題 (20)

複数のデータを昇順で保存する必要がある場合、配列とリスト構造で扱う場合の利点と欠点を説明せよ。

ただし、(1) 処理時間のオーダーと、(2) メモリの使用量、について説明してあること。