

設問より5つを選択し解答せよ。

## 1 リストの動作トレース

下線部は、後述の設問にて利用する。

```
#include <stdio.h>
struct List {
    int data ;
    struct List* next ;
} ;

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

struct List* stack = NULL ;
void push( int x ) {
    stack = cons( x , stack ) ;
} ~~~~~D

int pop() {
    struct List* del = stack ;
    int ret = stack->data ;
    stack = stack->next ;
    free( del ) ;
    return ret ;
}

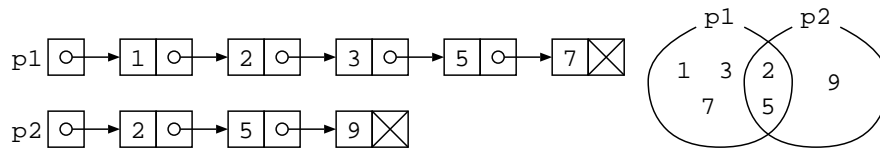
void main() {
    push( 1 ) ; push( 22 ) ; push( 333 ) ;
    printf( "%d\n" , pop() ) ; /* (A) */
    printf( "%d\n" , pop() ) ; /* (B) */
    printf( "%d\n" , pop() ) ; /* (C) */
} ~~~~~E
```

- 上記のプログラムの (A) が終了した時点で、stack の先に生成されているリスト構造のイメージ図を示せ。
- プログラムの (C) が終了した時点で、表示される内容を示せ。

## 2 リストによる集合処理

前設問と同じ整数データのリスト構造を使って、2つの整数集合  $A, B$  の差集合  $A - B$  を求めるプログラムを作成したい。

例えば下記のようなデータの場合、 $p1 - p2 = \{1, 3, 7\}$  が求まることを期待する。



```
struct List* diff( struct List* p1 , struct List* p2 ) {  
{  
    /* この関数内部を作成すること */  
}
```

### 3 配列とリスト処理の比較

複数の数値データを管理するためのプログラムを、配列とリスト構造によりそれぞれ記述した。イメージ図を参考に、配列にデータを1件追加するプログラム `array_insert(n,x)` と、リスト構造にデータを1件追加する `list_insert(p,x)` を記述せよ。

```
#include <stdio.h>
int array[ 100 ] ; /* 数値データ */
int size ; /* 配列内のデータ件数 */

void array_insert( int n , int x ) {

    /* この部分を記述せよ */

}

void array_remove( int n ) {
    int i ;
    for( i = n + 1 ; i < size ; i++ )
        array[ i ] = array[ i + 1 ] ;
    --size ;
}

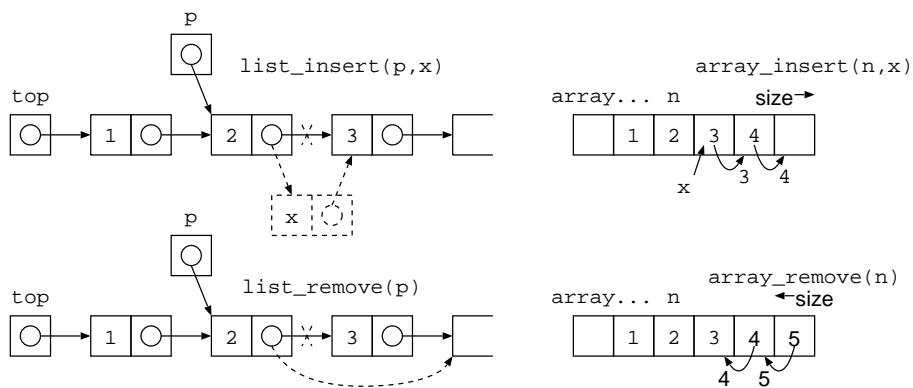
struct List { /* リスト構造の宣言 */
    int data ;
    struct List* next ;
} ;
struct List* top ; /* リストの先頭 */

void list_insert( struct List* p , int x ) {

    /* この部分を記述せよ */

}

void list_remove( struct List* p ) {
    struct List* d = p->next ;
    p->next = d->next ;
    free( d ) ;
}
```



#### 4 処理速度のオーダーの説明

前設問のarray\_insert(n,x),list\_insert(p,x) の処理速度のオーダーが何になるか、それぞれ具体的な説明を交えて述べよ。

## 5 プログラムの型

最初の設問での下線部の型を述べよ。

(A) \_\_\_\_\_

(B) \_\_\_\_\_

(C) \_\_\_\_\_

(D) \_\_\_\_\_

(E) \_\_\_\_\_

## 6 リスト処理プログラム

名前 `name` と誕生日 `month, day` からなるリスト構造において、日付の範囲に含まれるデータの名前を表示する関数 `print_range()` を作成せよ。

これに加え、リスト構造の宣言も行うこと。

```
struct NameBirthList {
    /* リスト構造の宣言を記述せよ */

};
```

```
void print_range( struct NameBirthList* p ,
                  int sm , int sd , int em , int ed )
{ /* 日付範囲に含まれる要素の名前を表示せよ */

}

void main() {
    struct NameBirthList* top ;
    top = .... ; /* リスト構造を生成する処理... */

    printf( "乙女座の人は...\n" ) ;
    print_range( top , 8 , 23 , 9 , 22 ) ;
    printf( "です。 \n" ) ;
}
```

