

以下の6つの設問の中から、5つを選んで解答せよ。

1 2分木問題 (20)

```

struct Tree {
    int data ;
    struct Tree* left ;
    struct Tree* right ;
};

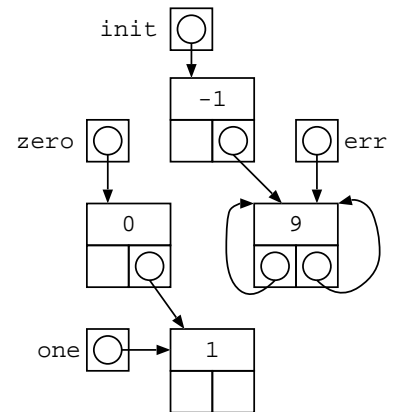
struct Tree* tcons( int x ,
                    struct Tree* L , struct Tree* R ) {
    struct Tree* ans ;
    ans = (struct Tree*)malloc( sizeof( struct Tree ) ) ;
    if ( ans != NULL ) {
        ans->data = x ;
        ans->left = L ;
        ans->right = R ;
    }
    return ans ;
}
    
```

```

void main() {
    struct Tree* err = tcons( 9 , NULL , NULL ) ;
    err->left = err ;
    err->right = err ;
    struct Tree* one = tcons( 1 , err , NULL ) ;
    one->right = one ;
    struct Tree* zero = tcons( 0 , NULL , one ) ;
    zero->left = zero ;
    struct Tree* init = tcons( -1 , zero , err ) ;

    struct Tree* p = init ;
    char str[] = "00110" ;

    for( int i = 0 ; str[i] != '\0' ; i++ ) {
        printf( "%d," , p->data ) ;
        if ( str[i] == '0' )
            p = p->left ;
        else if ( str[i] == '1' )
            p = p->right ;
        else
            p = err ;
    }
    printf( "\n(%d)\n" , p->data ) ;
}
    
```



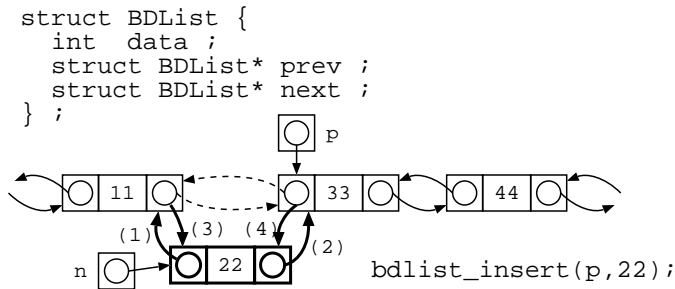
1. 上記のプログラムで生成されるデータ構造のイメージ図を完成させよ。(12)
2. 上記のプログラムの実行結果を答えよ。(8)

2 正規表現と説明問題

1. 前設問のプログラムで、入力として様々な `str[]` の文字を与えるとする。終了時に `p` が `one` を指している場合、`str` の文字列はどのようなものが正規表現で表現せよ。(10)
2. インタプリタとコンパイラの欠点を3つ答えよ。(10)

3 双方向リスト

以下に示すような双方向リストにおいて、指定した要素 p の前に指定した値を挿入する関数 `bdlist_insert(p , val)` の空白部を埋めよ。



```

void bdlist_insert( _____ p , int val ) {
    struct BDList* n ; ~~~~~(A)2

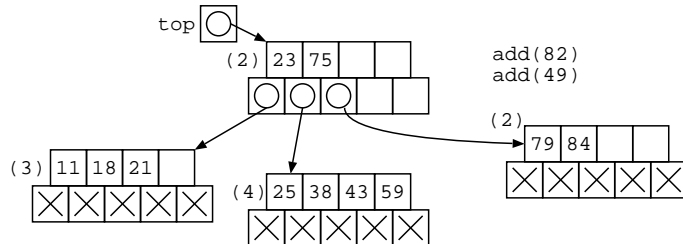
    n = ( _____ )malloc( _____ ) ;
        ~~~~~(B)3 ~~~~~(C)3

    if ( n != NULL ) {
        n->prev = _____ // (1)
        n->next = _____ // (2)
        ~~~~~(E)3 // (3)
        ~~~~~(F)3 // (4)
    } ~~~~~(G)3
}
    
```

`n->data = val ;`

4 説明問題

以下のような図で示される B 木において、データ 82, 49 を追加した場合、B 木にはどのようにデータが追加されるのか、説明を交えながら追加後の状態を図示せよ。



5 型の理解と動作トレース

2分探索木の登録処理と、そのデータを扱う関数 print() を作成した。

```
struct Tree* top = NULL ;
void tree_add( int val ) {
    struct Tree** tail = &top ;
    while( *tail != NULL ) {
        if ( (*tail)->data == val )
            break ;
        else if ( (*tail)->data > val )
            tail = &( (*tail)->left ) ;
        else
            tail = &( (*tail)->right ) ;
    }
    if ( *tail == NULL )
        *tail = tcons( val , NULL , NULL ) ;
}

void print( struct Tree* p ) {
    if ( p != NULL ) {
        printf( "(" ) ;
        print( p->left ) ;
        printf( "%d" , p->data ) ;
        print( p->right ) ;
        printf( ")" ) ;
    }
}

void main() {
    tree_add( 50 ) ;
    tree_add( 25 ) ;
    tree_add( 75 ) ;
    print( top ) ;
}
```

[設問 2] プログラム中の下線部の型を答えよ。(2x5)

(A) _____ (B) _____ (C) _____
(D) _____ (E) _____

6 動作トレース

```
struct Tree* tree_add2( struct Tree* p , int val ) {
    if ( p == NULL )
        return tcons( val , NULL , NULL ) ;
    else if ( p->data < val ) // 注意せよ
        p->left = tree_add2( p->left , val ) ;
    else
        p->right = tree_add2( p->right , val ) ;
    return p ;
}

void main() {
    struct Tree* top = NULL ;
    top = tree_add2( top , 50 ) ;
    top = tree_add2( top , 25 ) ;
    top = tree_add2( top , 75 ) ;
    top = tree_add2( top , 80 ) ;
    print( top ) ;
}
```

[設問]
このプログラムの実行結果
を答えよ。(20)
