



6つの選択問題より、5個を選んで回答せよ。

1 処理トレース

```

struct Tree {
    int data ;
    struct Tree* left ;
    struct Tree* right ;
};
struct Tree* top = NULL ;

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

struct Tree* tcons( int d , struct Tree* l , struct Tree* r ) {
    struct Tree* n ;
    n = (struct Tree*)malloc( sizeof( struct Tree ) ) ;
    if ( n != NULL ) {
        n->data = d ;
        n->left = l ;
        n->right = r ;
    }
    return n ;
}

void main() {
    entry( 50 ) ;
    entry( 75 ) ;
    entry( 25 ) ;
    entry( 30 ) ;
}

```

設問 1:main 終了後のデータ構造のイメージ図を示しなさい。(10)

設問 2:下線部 (A)-(E) の型を答えよ。(10)
(A) _____, (B) _____, (C) _____
(D) _____, (E) _____

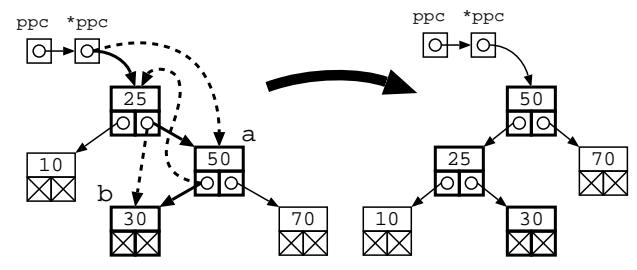
2 AVL 木の回転処理

不均一な 2 分木が生成されてしまった場合、枝の先のデータの件数に応じてポインタの繋ぎ替えを行う。左回転と呼ばれる処理では、下図左の直線で示されるポインタを点線のように繋ぎ替え、最終的に右のようにデータが配置されるようにする。ふさわしい式を (A)-(E) に記載せよ。

```

void rot_left( struct Tree** ppc ) {
    struct Tree* a = _____ ; (A)
    struct Tree* b = _____ ; (B)
    (*ppc)->right = _____ ; (C)
    _____ ; (D)
    *ppc = _____ ; (E)
}

```



3 状態遷移

```
void main() {
    struct Tree* a = tcons( 'a' , NULL , NULL ) ;
    struct Tree* b = tcons( 'b' , NULL , NULL ) ;
    struct Tree* c = tcons( 'c' , NULL , NULL ) ;
    struct Tree* p ;
    int          i ;

    a->left  = b ;
    a->right = c ;
    b->left  = a ;
    b->right = b ;
    c->left  = b ;
    c->right = c ;
    /* (A) */
    p = a ;
    for( i = 0 ; i < 4 ; i++ ) {
        if ( array[i] == 0 )
            p = p->left ;
        else
            p = p->right ;
        printf( "(%d)%c" , i , p->data ) ;
    }
}
```

設問 1: (A) の時点で生成されているデータ構造のイメージ図を示せ。(8)

設問 2: このプログラムの実行結果を答えよ。(3x4)

(0)_____ (1)_____ (2)_____ (3)_____

4 2分木による式の表現

式 $7+(a-2)$ のような、1 桁の数値 (0-9)、1 文字の変数名 (a-z)、演算子 (+,-) で構成される式を 2 分木で表現するものとする。下のような関数 eval() を実行したら、17 と表示したい。

途中まで記述された関数 eval() の (A),(B) を埋め、残りの部分を作成せよ。

```
printf( "%d\n" ,
        eval( tcons( '+' ,
                    tcons( '7',NULL,NULL) ,
                    tcons( '-', tcons( 'a',NULL,NULL) , tcons( '2',NULL,NULL) ) ) ) ) ;
-----
int var[ 26 ] = { 12 , 34 , 56 } ; // 変数 A-Z の値の入った配列とする

int eval( struct Tree* p ) {
    if ( p->data >= 'a' && p->data <= 'z' )
        return var[ _____ ] ;
    else if ( p->data >= '0' && p->data <= '9' )
        return _____ (B)
```

5 ハッシュ法

名前と電話番号のデータをチェイン法で保存するプログラムで、データを登録・検索を行う関数の下線部を埋めよ。(2x10)

```
#define HSIZE 10
struct PNList {
    int phone ;
    char name[ 10 ] ;
    struct PNList* next ;
} *hash[ HSIZE ] ;

int hash_func( int tel ) {
    return tel % HSIZE ;
}

void entry( int ph , char nm[] ) {
    int idx = _____ ; (D)
}

struct PNList* pncons( int ph , char nm[] ,
                       struct PNList* nx )
{
    struct PNList* n = _____ (A)
    if ( _____ ) (B) {
        n->phone = ph ;
        _____ (C)
        n->next = nx ;
    }
    return n ;
}

char* search( int ph ) {
    int idx = _____ (F)
    struct PNList* p ;
    for( p = _____ ; (G) _____ ; (H) _____ ) (I)
        if ( _____ ) (J)
            return p->name ;
    return NULL ;
}

void main() {
    entry( 272925 , "tohru" ) ;
    entry( 272546 , "saitoh" ) ;
    entry( 123456 , "foobar" ) ;
}
```

6 説明問題

B木について、(1) データ構造のイメージ図、(2) データの検索方法、(3) 不均一な構造にならないための工夫を交えて、具体的に説明せよ。(7+7+6)