

以下の6つの設問の中から、5つを選んで解答せよ。

## 1 基礎問題と再帰呼出し

```

struct Tree {          | void print( int idx ) {
    int data ;         |     if ( idx >= 0 ) {
    int left ;         |         print( table[ idx ].left ) ;
    int right ;        |         printf( "%d " , table[ idx ].data ) ;
} table[ 4 ] = {       |         print( table[ idx ].right ) ;
    { 11 , 3 , 2 } , // 0 |     }
    { 22 ,-1 ,-1 } , // 1 | }
    { 33 , 1 , 3 } , // 2 | // プログラムの実行結果
    { 44 ,-1 ,-1 } , // 3 | void main() { // 答えよ。(4x5)
} ;                    |     print( 0 ) ; // 再帰呼出しの状況が解る図
                        | } // があれば中間点を考慮する。

```

## 2 コンパイラに関する用語穴埋め

C 言語が、コンパイラによって機械語を生成するまでの処理の流れで、[\*] データと (\*) 処理を交互に記載した。下線部にふさわしい処理やデータ形式の名前を答えよ。(4x5)

[A] "#行"を含む C 言語のソース

↓ (A) \_\_\_\_\_

[B] "#行"の無い C 言語のソース

↓ (B) \_\_\_\_\_

[C] 演算子記号や名前のようなトークンに分割されたもの

↓ (C) \_\_\_\_\_

[D] 文法に構造を表す木構造データ

↓ (D) 命令生成処理

[E] \_\_\_\_\_

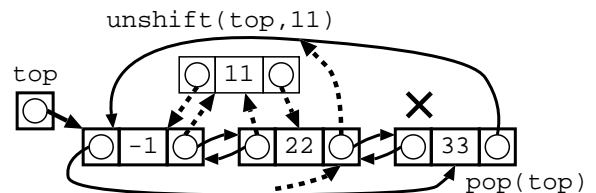
↓ (E) \_\_\_\_\_ ← ライブラリ

[F] 機械語

### 3 双方向リストの記述問題 (20)

双方向リストの先頭に要素を追加する `unshift()`、  
末尾の要素を取り出す `pop()` を、右図を参考にプ  
ログラムを作成せよ。

```
struct Deque {
    int data ;
    struct Deque* prev ;
    struct Deque* next ;
};
```



```
struct Deque* dq_cons( int x , struct Deque* p , struct Deque* n ) {
    struct Deque* ans = (struct Deque*)malloc( sizeof( struct Deque ) );

    if ( ans != NULL ) {
        ans->data = x ;
        ans->prev = p ;
        ans->next = n ;
    }
    return ans ;
}

void main() {
    struct Deque* top ;
    top = dq_cons( -1 , NULL ,
                  dq_cons( 22 , NULL ,
                          dq_cons( 33 , NULL , NULL ) ) ) ;
    top->next->prev = top ;
    top->next->next->prev = top->next ;
    top->next->next->next = top ;
    top->prev = top->next->next ;
    unshift( top , 11 ) ;
    printf( "%d\n" , pop( top ) ) ;
}
```

```
void unshift( struct Deque* p , int x ) {
    int pop( struct Deque* p ) {
```

### 4 説明問題 (20)

以下の2つの説明問題から1つを選んで回答せよ。

1. B木の構造について説明し、データの追加処理でどのようなことを行うか説明せよ。
2. 双方向リストの設問で、データ先頭の-1の入った要素は、どのような目的で使われているか説明し、この手法の名称を答えよ。

## 5 プログラム記述問題

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

void add( int x ) {
    struct Tree** tail = &top ;
    while( _____ ) {
        if ( _____ )
            break ;
        else if ( (*tail)->data > x )
            tail = _____ ;
        else _____(F)
            tail = &( (*tail)->right ) ;
    }
    if ( *tail == NULL )
        *tail = tcons( x , NULL , NULL ) ;
}
void main() {
    add( 50 ) ; add( 20 ) ; add( 70 ) ; add( 30 ) ;
}
```

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

[1] (D),(E),F) にふさわしい処理を答えよ。

(D) \_\_\_\_\_ (4)  
(E) \_\_\_\_\_ (4)  
(F) \_\_\_\_\_ (4)

[2] (A),(C),(H) の型を答えよ。

(A) \_\_\_\_\_ (3)  
(C) \_\_\_\_\_ (2)  
(H) \_\_\_\_\_ (3)

## 6 説明問題 (20)

1. 前設問で (B) の判定を行う理由を答えよ。(6)
2. 2分木で、検索時間に関する欠点を1つ答えよ。(6)
3. 上記の問題点を解決するための方法の名称とその方法を簡単に説明せよ。(3+5)