5 つの設問の中から、4 つを選んで回答せよ。

1 動作トレース (25)

C 言語による再帰処理を用いた 2 分探索法のプログラム `find_rec()` について、以下のプログラムで表示される内容を答えよ。

```
int find_rec( int a[] , int key , int left , int right ) {  
    printf( "%d %d\n" , left , right ) ;  
    if ( left == right ) {  
        return -1 ;  
    } else {  
        int mid = (left + right) / 2 ;  
        if ( a[ mid ] == key )  
            return mid ;  
        else if ( a[ mid ] < key )  
            return find_rec( a,key,mid+1,right ) ;  
        else  
            return find_rec( a,key,left,mid ) ;  
    }  
}  
  
int b[] = { 11 , 22 , 33 , 44 } ;  
  
int main() {  
    printf( "%d\n" , find_rec( b , 33 , 0 , 4 ) ) ;  
    printf( "%d\n" , find_rec( b , 45 , 0 , 4 ) ) ;  
    return 0 ;  
}
```

```
int find_loop( int a[] , int key , int left , int right ) {  
    while( left != right ) {  
        printf( "%d %d\n" , left , right ) ;  
        int mid = (left + right) / 2 ;  
        if ( a[ mid ] == key )  
            return mid ;  
  
        else if ( _____ ) (A)  
            left = _____ ;  
        else _____ (B)  
            right = mid ;  
    }  
    return _____ (C)  
}
```

設問 (1) 実行結果を答えよ。(10)

```
-----  
-----  
-----  
-----  
-----
```

設問 (2) `find_rec()` と同じ結果が得られる `find_loop()` を作りたい。
プログラム中の下線部分にふさわしい処理を答えよ。(5x3)

2 処理時間とオーダー (25)

1. 前問 1 のような 2 分探索法の処理時間を、対象データ件数 N としてオーダー記法で答えよ。(5)
2. 処理時間の一般式が $T(N) = T_\alpha + T_\beta N^2 + T_\gamma N \log N$ で示されるアルゴリズムがあった場合、処理時間をオーダー記法で答えよ。(5)
3. 処理時間のオーダーが $O(N^2)$ で示されるアルゴリズムで、データが $N=50$ 件で 1μ 秒であった場合、データ $N=1000$ 件では何秒かかるか答えよ。(7)
4. 処理時間のオーダーが $O(2^{\frac{N}{2}})$ で示されるアルゴリズムで、データが $N=10$ 件で 1μ 秒であった場合、データ $N=30$ 件では何秒かかるか答えよ。(8)

3 プログラム穴埋め (25)

C 言語で、配列の要素を昇順に並べ替える下記の選択法のプログラムの、下線部分にふさわしい処理を答えよ。また、下記のプログラムで処理 (F) `sort()` の呼び出しの中で処理 `swap()` は何回呼び出されるか答えよ。

```
void swap( _____ pa , int* pb ) { // ポインタで示す場所の
    _____(A)3 // データを入れ替え
    int tmp = *pa ;
    *pa = *pb ;
} _____(B)4

void sort( int a[] , int size ) { // a[]: 配列,size: データ数
    for( int i = 0 ; i < _____ ; i++ ) {
        int min = i ;
        for( int j = i + 1 ; j < size ; _____ ) {
            if ( _____ ) {
                min = j ;
            }
            if ( min != i )
                swap( & a[i] , _____ ) ;
        }
    }
}

// swap() の呼び出し回数 :
int main() { _____(G)4
    int array[] = { 11 , 44 , 33 , 55 , 22 } ;
    sort( array , 5 ) ; /* (F) */
    for( int i = 0 ; i < 5 ; i++ )
        printf( "%d\n" , array[ i ] ) ; // 11, 22, 33, 44, 55
    return 0 ;
}
```

4 説明問題 (13+12)

1. Java における関数呼び出しで実引数と仮引数の値の受け渡し方で、値渡しと参照渡しの違いを、ポインタやイメージ図を交えながら、副作用の有無も交えて説明せよ。
2. プログラム作成におけるトレードオフの関係において、メモリを大量に消費するプログラムは、処理速度にどういった影響を及ぼす可能性があるか具体的な現象を交えながら説明せよ。

5 プログラム作成問題 (25)

Java により商品情報 (Item) の一覧 (item_list) と、購入情報 (Buy) の一覧 (buy_list) が与えられたら、購入総額を求める関数 total_price() を作成せよ。

商品情報は (商品 ID, 商品名, 商品単価) で構成され、購入情報は (購入する商品 ID, 購入個数) から構成される。

```
class Item {
    int id ;
    String name ;
    int price ;
    Item( int i , String n , int p ) {
        this.id = i ;
        this.name = n ;
        this.price = p ;
    }
};

class Buy {
    int id ;
    int count ;
    Buy( int i , int n ) {
        this.id = i ;
        this.count = n ;
    }
};

public class Main {
    static Item[] item_list = {
        new Item( 1010 , "orange" , 50 ) ,
        new Item( 1020 , "apple" , 100 ) ,
        new Item( 1022 , "pineapple" , 1000 ) ,
    };
    static Buy[] buy_list = {
        new Buy( 1010 , 5 ) ,
        new Buy( 1020 , 3 ) ,
        new Buy( 1022 , 1 ) ,
    };
    public static void main( String[] args ) {
        System.out.println( total_price( item_list , buy_list ) );
    }
    static int total_price( Item[] i_list , Buy[] b_list ) {
        // この部分を作成せよ
    }
}
```

同様のプログラムを C 言語で作成したのもも以下に示す。回答は Java, C のいずれかで回答せよ。

```
struct Item {
    int id ;
    char name[ 20 ] ;
    int price ;
};

struct Item item_list[] = {
    { 1010 , "orange" , 50 } ,
    { 1020 , "apple" , 100 } ,
    { 1022 , "pineapple" , 1000 } ,
};

struct Buy {
    int id ;
    int count ;
};

struct Buy buy_list[] = {
    { 1010 , 5 } ,
    { 1020 , 3 } ,
    { 1022 , 1 } ,
};

int total_price( struct Item item_list[] , int item_size ,
                struct Buy buy_list[] , int buy_size ) {
    // この部分を作成せよ
}

int main() {
    printf( "%d\n" , total_price( item_list , 3 , buy_list , 3 ) );
    return 0 ;
}
```