

1 処理速度の予測 (× 25)

1. 以下に示す処理時間の一般式より、処理のオーダを答えよ。(2 × 5)

(a) $T(N) = T_\alpha + T_\beta \sqrt{N} + T_\gamma 1.5^N$

(b) $T(N) = T_A + T_B N^{1.5} + T_C N \log N$

2. あるプログラム A の処理時間のオーダが、 $O(N^2 \log N)$ であった。このプログラムのデータ 100 件の時の処理時間が 10msec であった。このプログラムのデータ数が 1000 件の時の処理時間を求めよ。(× 7)

3. プログラム A と同じ処理を別なアルゴリズム B (処理オーダ $O(N \log N)$) で作ったら、データ 100 件で処理時間が 200 msec であった。

アルゴリズム B の方が速く終わるのは、データ件数が何件以上か求めよ。(× 8)

2 説明問題 (選択問題 × 25)

以下の 3 つの中から 2 つを選んで、説明せよ。(2 × 12,2 問完答で +1)

1. アルゴリズムを選択する上での 3 つの点をあげて、そのトレードオフの関係を説明せよ。
2. malloc などの動的メモリを使う処理で free を忘れてメモリが不足すると、一般的に処理速度が遅くなる。どうしてこういった現象が起こるのか、説明せよ。
3. ハノイの塔の処理ステップ $H(N)$ の一般式を求める時に必要な、再帰方程式を示せ。ただし、その再帰方程式を立てる際の意味の説明を具体的に行うこと。一般式の予測や証明は不要。

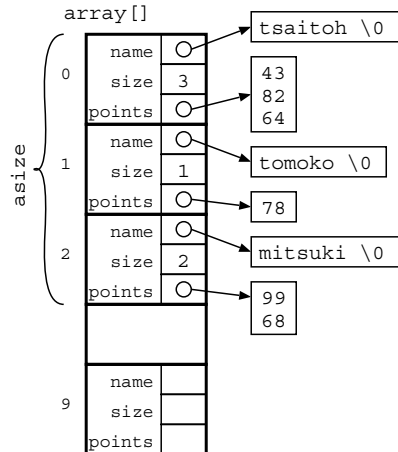
4 動的メモリ管理の穴埋め (× 25)

下に示すような、名前と点数のデータが与えられる。点数のデータは、20件未満で末尾の目印として、最後に-1が入力する。このデータを入力して覚えるプログラムを以下のように作った。下線部を埋めてプログラムを完成させよ。

```
tsaitoh 43 82 64 -1
tomoko 78 -1
mitsuki 99 68 -1
```

```
struct NamePoints {
    char* name ;
    int size ;
    int* points ;
};

void printNamePoints( struct NamePoints* p ) {
    int i ;
    printf( "%s:" , p->name ) ;
    for( i = 0 ; i < p->size ; i++ )
        printf( "%d " , p->points[ i ] ) ;
    printf( "\n" ) ;
}
```



```
void main() {
    char tmp_name[ 100 ] ;
    int tmp_points[ 20 ] ;
    struct NamePoints array[ 10 ] ;
    int asize, i, j ;
    for( asize = 0 ; scanf( "%s" , tmp_name ) == 1 ; asize++ ) {
        for( i = 0 ; i < 20 ; i++ ) {
            if ( scanf( "%d" , &tmp_points[ i ] ) != 1 || tmp_points[ i ] < 0 )
                break ;
        }
        array[ asize ].name = ~~~~~(A)
        if ( ~~~~~(B) )
            strcpy( array[ asize ].name , tmp_name ) ;
        array[ asize ].size = ~~~~~(C)
        array[ asize ].points = ~~~~~(D)
        if ( array[ asize ].points != NULL ) {
            for( j = 0 ; j < i ; j++ ) {
                ~~~~~(E)
            }
        }
    }
    for( i = 0 ; i < asize ; i++ ) {
        printNamePoints( ~~~~~(F) ) ;
        if ( ~~~~~(G) )
            free( array[ asize ].name ) ;
        if ( array[ asize ].points != NULL )
            ~~~~~(H)
    }
}
```