

1 動作トレース (20)

```
// 設問 (1-0)
int array[] = { 1, 2, 3, 4, 5, 6, 7 };

int sum( int a[] , int L , int R ) {
    if ( R - L == 1 ) {
        return a[ L ];
    } else {
        int M = (L + R) / 2 ;
        int sL = sum( a , L , M ) ;
        int sR = sum( a , M + 1 , R ) ;
        return sL + a[ M ] + sR ;
    }
}

int main() {
    printf( "%d\n" , sum( array, 0, 7 ) ) ;
    return 0 ;
}

// 設問 (1-1)
int array[] = { 1 , 2 , 3 , 4 } ;

int sum( int a[] , int size , int i ) {
    if ( i >= size ) {
        return 0 ;
    } else {
        int sL = sum( a , size , 2 * i + 1 ) ;
        int sR = sum( a , size , 2 * i + 2 ) ;
        return a[i] + sL + sR ;
    }
}

int main() {
    printf( "%d\n" , sum( array , 4 , 0 ) ) ;
    return 0 ;
}
```

上記プログラムの指定された (1-0),(1-1) のいずれかの実行結果を答えよ。(8+12)
ただし、このプログラムで再帰がどのように発生しているか、(1-0) は L,R の値,(1-1) は i の値に注目し、具体的な引数の値などを図と交えて説明すること。

2 オータダ問題 (20)

以下の、処理時間と再起方程式の問題について、指定されたいずれかについて答えよ。(10+10)
処理時間の見積もりでは、解くために必要な式と説明を交えて、答えること。
また、再帰方程式は、一般式まで求めなくて良い。式中の時間に関する定数がある場合は、それがプログラムの何の処理時間か説明すること。

設問 (2-0)

1. ハノイの塔でディスクの枚数を N とした時、積み上げられた $N = 10$ 枚をとりなりに移動するのに 1 時間を要した。ディスクの枚数が $N = 20$ の時、処理にかかる時間を予想せよ。
ただし、近似として $2^{10} \simeq 1000$ を利用して良い。
2. 以下のプログラムの処理時間を再帰方程式で示せ。

```
int add( int n , int m ) {
    if ( n == 0 ) {
        return m ;
    } else {
        return add( n - 1 , m + 1 ) ;
    }
}
// printf( "%d" , add( 5 , 10 ) ) ;
```

設問 (2-1)

1. マージソートでデータを並び替える処理処理でデータ件数を N とした時、 $N = 100$ で $50[\mu \text{ sec}]$ の時間がかかった。 $N = 1000$ の時に要する処理時間を予想せよ。
2. 以下のプログラムの処理時間を再帰方程式で示せ。

```
int mul( int n , int m ) {
    if ( n <= 1 ) {
        return n * m ;
    } else {
        return ( n % 2 ) * m
            + mul( n / 2 , m * 2 ) ;
    }
}
// printf( "%d" , mul( 5 , 10 ) ) ;
```

3 オーダ問題 (20)

以下のプログラムについて、(1) 実行結果と、(2) 指定した式について式の各部分の型について説明せよ。回答は、設問 (3-0) の場合は (A),(C),(E),(G) について、設問 (3-1) の場合は (B),(D),(F),(H) について答えること。(3x4 + 8)

```
struct Data {
    char nm[ 10 ] ;
    int pt[ 3 ] ;
    double avg ;
} ;

struct Data stu[ 3 ] = {
    { "aoyama" , { 84 , 93 , -1 } , 88.50 } ,
    { "saitoh" , { 72 , 62 , 48 } , 60.67 } ,
    { "sakamoto" , { -1 , -1 , 81 } , 81.00 } ,
} ;

int main() {
    printf( "%c\n" , stu[ 2 ].nm[ 2 ] ) ;
    printf( "%d\n" , stu[ 1 ].pt[ 1 ] ) ;

    printf( "%lf\n" , stu->avg ) ;
    printf( "%d\n" , *(stu[ 1 ].pt) ) ;
    struct Data* p = stu + 2 ;
    printf( "%d\n" , *(p->pt) ) ;
    printf( "%s\n" , p[ -1 ].nm ) ;

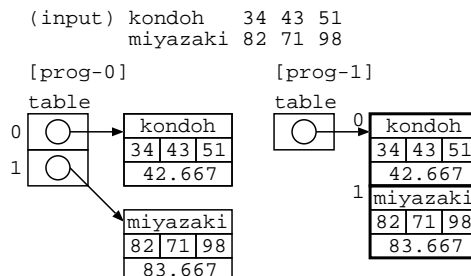
    printf( "%d\n" , (*(p - 1)).pt ) ;
    printf( "%d\n" , *((p - 2)->pt + 1) ) ;
    return 0 ;
}
```

// 設問 (3-0)
// 設問 (3-1)
// (A)-----
// (B)-----
// (C)-----
// (D)-----
// (E)-----
// (F)-----
// (G)-----
// (H)-----
// (G) の型の説明
// (H) の型の説明

4 プログラム作成問題 (20)

前問題について、以下の図のようなイメージ図となるように、変数を宣言し入力したデータを格納せよ。ただし、データは名前と3科目のテストが与えられ、参加していないテストの欄には-1が入っているものとし、avgには参加したテストの平均点(小数点以下も求まるように)を格納せよ。与えられるデータは、必ず2件とし、ポインタで示される領域は malloc() でヒープメモリ上に確保すること。構造体の型の宣言や確保したメモリの開放などは、省略して良い。

設問 (4-0) は [prog-0]、設問 (4-1) は [prog-1] の図を回答すること。



5 説明問題 (20)

以下の説明問題について回答せよ。

設問 (5-0)

1. 末尾再帰呼び出しについて、プログラム例などを交えて説明せよ。
2. 処理速度を優先したら、メモリを大量に消費する事例について具体例を交えながら説明せよ。

設問 (5-1)

1. LIFO について、処理の例などを交えて説明せよ。
2. 分かりやすさを優先したら、処理速度が遅くなる事例について具体例を交えながら説明せよ。

回答する問題について

設問の回答は、以下の値に応じて、選ぶこと。

例えば「出席番号 7 番」は、(1-1), (2-1), (3-1), (4-0), (5-0) について回答すること。

1. 問題 1 は、出席番号 % 2
2. 問題 2 は、(出席番号 / 2) % 2
3. 問題 3 は、(出席番号 / 4) % 2
4. 問題 4 は、(出席番号 / 8) % 2
5. 問題 5 は、(出席番号 / 16) % 2