

情報構造論 前期中間試験(出題: 齊藤) 番号:

4EI 2004/06/08,(1/3) 氏名:

2つの必須問題と、3つの選択問題の中から2つを選択し回答せよ。選択問題が3つとも回答されていれば、最大100点になるまで1/3で加点する。

## 1 再帰問題 (必須, × 30)

```
struct BirthDay { /* 誕生日の構造体 */
    int month , day ;
};

/* 誕生日の比較関数 */
/* p > (m/d) なら正 : p = (m/d) なら 0 : p < (m/d) なら負 */

int cmp_birthday( _____ p , int m , _____ d )
{   if ( p->month == m )           [A]
    return p->day - d ;
    else
        return _____ ;
}

struct BirthDay table[] = { /* 昇順に並べられている誕生日の配列 */
    { 2 , 7 } , { 4 , 22 } , { 7 , 14 } , { 9 , 18 }
};
int find1( struct BirthDay a[] , int l , int r , int bm , int bd )
{   /* 2分探索法でデータを探す */
    while( r-l > 0 ) {
        int m = (1 + r) / 2 ;
        int c = cmp_birthday( _____ , bm , bd ) ;
        if ( _____ )           (D)
            [E]
            return m ; /* 見つかった */
        else if ( c > 0 )
            r = m ;
        else
            l = m + 1 ;
    }
    return -1 ; /* 見つからない */
}
void main() {
    printf( "%d 番目で見つかった " , find1( table , 0 , 4 , 7 , 14 ) ) ;
}
```

### 1.1 穴埋め質問

- (D)に相応しい式を、以下の式の中から1つ選べ。  
(1)&a[m] , (2)a + r , (3)m + a , (4)&a[l]
- [A] ~ [E] に相応しいプログラムを埋めよ。

## 1.2 応用問題

前のプログラムを再帰呼び出しによって記述したプログラムを以下に示す。

```
int find2( struct Birthday a[] , int l , int r , int bm , int bd )
{
    int c ;
    if ( r <= 1 )
        return -1 ; /* 末端まで探した */
    if ( (c = cmp_birthday( &a[ l ] , bm , bd )) == 0 )
        return 1 ; /* 見つかった */
    else if ( c > 0 )
        return -1 ; /* 通り過ぎた */
    else
        return find2( a , l + 1 , r , bm , bd ) ; /* 再帰で探す */
}
void main() {
    printf( "%d 番目で見つかった " , find2( table , 0 , 4 , 7 , 14 ) ) ;
}
```

このプログラムの処理速度のオーダーを示せ。

ヒント：データが途中で見つからない場合を先に考えると判りやすい。

## 2 処理速度見積り問題 (必須, × 20)

データの並び替えを行う、クイックソートと最大値選択法によるプログラムを作成した。アルゴリズムの処理速度の一般式は、以下の2つで示される。

アルゴリズム	処理速度一般式
(1)	$T_1(N) = T_a N^2 + T_b N + T_c$ ( $T_a \sim T_f$ は定数項)
(2)	$T_2(N) = T_d N \log N + T_e N + T_f$

1. アルゴリズム 1,2 のどちらが、クイックソートの一般式か示し、それぞれの処理速度をオーダ記法で示せ。

2. クイックソートのプログラムの処理速度が、データ件数  $N = 100$  で、 $200[\text{msec}]$ であった場合、データ件数  $N$  が  $10000$  件では、どの程度の処理時間と予想されるか答えよ。導出する際には、具体的な式等を交えて回答すること。

### 3 説明問題

以下の3つの用語の中から2つについて説明を加えよ。

(1) LIFO , (2) メモリリーク, (3) alloca

#### 4 ヒープメモリの使い方

以下のプログラムでは、名前と年齢のデータをヒープメモリ上で保存し、並べ替え処理の後、表示・開放する。データを保存する処理部分が未完成となっているので、プログラムを完成させよ。

```
struct Person {          /* 名前と年齢の構造体 */
    char name[ 20 ] ;
    int  age ;
} ;
struct Person *table[ 20 ] ; /* ポインタでデータを保存 */
int          size = 0 ;      /* 入力したデータ件数   */

char nm[ 20 ] ;
int  ag , i , j , m ;

/* データの入力と領域確保 */
while( scanf( "%s %d" , nm , &ag ) == 2 ) {
    /* nm: 名前, ag: 年齢のデータを配列に保存する処理を記述せよ */
    /* この部分を下の余白部に記述すること */
}
/* ポインタのデータを年齢順で並び変える */
for( i = 0 ; i < size - 1 ; i++ ) {
    struct Person* p ;
    for( m = i , j = i + 1 ; j < size ; j++ ) {
        if ( table[ m ]->age > table[ j ]->age )
            m = j ;
    }
    p = table[ m ] ; table[ m ] = table[ i ] ; table[ i ] = p ;
}
/* データの表示とデータの開放 */
for( i = 0 ; i < size ; i++ ) {
    printf( "%s %d\n" , table[ i ]->name , table[ i ]->age ) ;
    free( table[ i ] ) ;
}
```

## 5 再帰処理

以下のプログラムの動作結果を示せ。

ヒント：2進数で考えると、計算の意味が見えてくる。

```
int foo( int x , int y ) /* % は余り、*/
{   if ( x == 0 )      /* / は整数除算で切捨て */
    return 0 ;
    else
        return y * ( x % 2 ) + foo( x / 2 , y * 2 ) ;
}
void main() {
    printf( "%d\n" , foo( 11 , 10 ) ) ;
    printf( "%d\n" , foo( 50 , 10 ) ) ;
}
```