

6 つの設問の中から、5 つを選んで回答せよ。

1 動作予測 (4 × 5)

以下のプログラムの実行結果を答えよ。

(再帰の状態が分かる図を交えてあれば、中間点なども考慮する)

```
int rec_max( int array[] , int L , int R ) | int a[] = { 23 , 52 , 21 , 39 , 80 } ;
{ | int main() {
  if ( R - L == 1 ) { |     printf( "%d\n" ,
    return array[ L ] ; |         rec_max( a , 0 , 5 ) ) ;
  } else { |     return 0 ;
    int M = (L + R) / 2 ; | }
    int lmx = rec_max( array , L , M ) ; | -----
    int rmx = rec_max( array , M , R ) ; | -----
    printf( "%d : %d\n" , lmx , rmx ) ; | -----
    if ( lmx > rmx ) | -----
      return lmx ; | -----
    else | -----
      return rmx ; | -----
  } | -----
} | -----
```

2 オータ記法 (20)

- あるプログラムの処理時間がデータ数 N とした時、 $T_f(N) = T_\alpha + T_\beta(1.62^N - 0.62^N)$ で示される場合、この処理時間をオーダ記法で答えよ。(5)
- あるプログラムの処理時間のオーダが、 $O(\sqrt{N} \times \log N)$ であった。データ件数 $N = 100$ 件で処理が 20msec であった場合、データ件数 $N = 10000$ 件の処理時間を予想せよ。(8)
- 前設問 1 の `rec_max()` の対象データ件数を N とした時、この処理時間 $T(N)$ にふさわしい再帰方程式を答えよ。何らかの処理時間を表す定数などは、 T_α, T_β, \dots などを用いること。(7)

3 説明問題 (10 × 2)

1. ヒープメモリを大量に使う処理で、free() を忘れると、処理速度にどういった影響がでるか、その理由を交えて説明せよ。
2. プログラムを作成するにあたって、トレードオフの関係にある3つの点をあげ、その点の相互間の影響を具体的な例を交えて説明せよ。(例えば3点がA,B,Cであった場合、A-B,B-C,C-Aの間の影響3つを説明すること)

4 穴埋め問題 (20)

入力した1行1単語で、同じ単語の頻度を数えるプログラムの下線部を埋めよ。

```
struct WordCount { | int wordsize = 0 ;
                   | char* word ;
                   | struct WordCount* table[ 100 ] ;
                   | int count ;
                   | char buff[ 1000 ] ;
};

int main() {
    while( wordsize < 100
           && fgets( buff , _____ , stdin ) != NULL ) {
        int i ;
        for( i = 0 ; i < wordsize ; i++ )
            if ( strcmp( buff , table[ i ]->word ) == 0 )
                break ; // strcmp(A,B) は文字列比較:同じ文字列なら 0 を返す
        if ( i < wordsize ) {
            _____ // 同じ単語あり:カウントアップ
            (B)3
        } else {
            table[ i ] = (struct WordCount*)malloc( _____ ) ;
            table[ i ]->word = strdup( buff ) ; _____ (C)3
            _____ (D)3
            _____ (E)3
        } _____ (F)3
    }
    for( int i = 0 ; i < wordsize ; i++ )
        printf( "%s %d\n" , _____ word , _____ count ) ;
    return 0 ; _____ (G)2
}
```

5 型の理解 (20)

```
struct YMD {          | struct NameBirthday {          | 以下のプログラムを
    int year ;        |     char      *name ;          | 実行した時に表示
    int month ;       |     struct YMD birthday ;     | される値を答えよ。
    int day ;         | } ;
} ;

struct NameBirthday table[ 3 ] = {
    { "Yuji Itadori" , { 2002 , 3 , 10 } } ,
    { "Megumi Fushiguro" , { 2001 , 12 , 22 } } ,
    { "Satoru Gojo" , { 1989 , 12 , 7 } } ,
} ;

void printBirthday( struct YMD* ymd ) {
    printf( "%d,%d/%d\n" ,
            ymd->year , ymd->month , ymd->day ) ;
}

int main() {
    struct NameBirthday *ptr[ 2 ] ;
    struct NameBirthday *p ;
    ptr[ 1 ] = &(amp;table[ 2 ]) ;
    ptr[ 0 ] = (struct NameBirthday*)malloc( sizeof( struct NameBirthday ) ) ;
    ptr[ 0 ]->name = (char*)malloc( 13 ) ;
    strcpy( ptr[ 0 ]->name , "Yuta Okkotsu" ) ;
    ptr[ 0 ]->birthday.year = 2001 ;
    ptr[ 0 ]->birthday.month = 3 ;
    ptr[ 0 ]->birthday.day = 7 ;                               // 回答欄

    printf( "%s\n" , table[1].name ) ;                          // -----(2)

    printf( "%d\n" , table->birthday.day ) ;                    // -----(3)
    p = table + 1 ;
    p-- ;
    printBirthday( &(amp;p->birthday) ) ;                        // -----(3)

    printf( "%c\n" , table[0].name[1] ) ;                       // -----(3)

    printf( "%s\n" , table[ 2 ].name + 7 ) ;                    // -----(3)

    printf( "%d\n" , (*ptr[1]).birthday.month ) ;              // -----(3)

    printf( "%c\n" , *(ptr[1]->name + 1) + 1 ) ;              // -----(3)
    return 0 ;
}
```

6 レポート課題相当 (20)

名前と3教科のテストのデータが1行毎に入力として与えられる。テストを受験しなかった教科の所には、-1が入っている。データ件数は最大100件とし、名前と点数が正しく読めなくなったら終わりとする。

回答欄は、裏を使うこと。

1. このデータを保存するための構造体を宣言し
2. 入力したデータを一旦配列に全件保存した後
3. それぞれの人の合計点数を表示せよ。

```
[input]                | [output]
saitoh 72 65 90        | saitoh 227
tomoko 82 -1 60        | tomoko 142
mitsuki -1 0 100       | mitsuki 100
入力終了
```