

以下の5つの設問の中から、4つを選んで解答せよ。

1 分割統治法と関数ポインタ (25)

```
int add( int x , int y ) { | [設問]
    return x + y ;        | 以下のプログラムの実行結果を答えよ。
}                          | ただし、答えが求まっていく経過を
                            | 分かりやすく示すこと。
int mul( int x , int y ) { |
    return x * y ;       | int a[] = { 2 , 3 } ;
}                          | int b[] = { 40 , 30 , 20 , 10 } ;
int map( int ar[] , int left , int right , int (*func)( int , int ) ) {
    if ( right - left == 1 ) {
        return ar[ left ] ;
    } else {
        int mid = (left + right) / 2 ;
        int lf = map( ar , left , mid , func ) ;
        int rf = map( ar , mid , right , func ) ;
        return (*func)( lf , rf ) ;
    }
}
                                                                    // 解答欄
int main() {
    printf( "%d\n" , map( a , 0 , 2 , mul ) ) ; // _____ (5) 過程 (5)
    printf( "%d\n" , map( b , 0 , 4 , add ) ) ; // _____ (5) 過程 (10)
    return 0 ;
}
```

2 アルゴリズムの効率 (25)

1. 処理時間のオーダが $O(N\sqrt{N})$ で示されるアルゴリズムがある。このアルゴリズムで、データ件数 $N = 100$ 件で処理時間が 10 [msec] かかったとする。データ件数が $N = 900$ 件では、処理時間はどの程度となるか答えよ。(9)
2. 以下のデータ構造の処理においてデータ件数 N に対する、処理時間にふさわしいオーダ式を一覧より選んで答えよ。(4x4)
 - (a) 一般的な2分探索木の中から特定の値を探す。
 - (b) デタラメな順序の配列の要素を最大選択法で並び替える。
 - (c) 線形リスト(単純なリスト)の全合計を求める。
 - (d) Dequeの末尾に1件要素を追加する。

一覧: $O(1), O(\log N), O(N), O(1.7^N), O(2^N), O(N\sqrt{N}), O(N \log N), O(N^2)$

3 参照カウンタ法 (25)

```
struct ListRC {          | 参照カウンタを用いた下記のプログラムで
    int          refc ;   | (1) (A) において a,b,c の先のデータ構造の
    int          data ;  | イメージ図を答えよ。 (13)
    struct ListRC* next ; | (2) main() の中の free_list() を呼び出し
} ;                      | 際に表示される内容を答えよ。(12)

struct ListRC* cons( int x , struct ListRC* n ) {
    struct ListRC* ans = (struct ListRC*)malloc( sizeof( struct ListRC ) ) ;
    if ( ans != NULL ) {
        ans->refc = 1 ;
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
}

struct ListRC* copy(
    struct ListRC* p ) {
    p->refc++ ;
    return p ;
}

void free_list( struct ListRC* p ) {
    while( p != NULL ) {
        p->refc-- ;
        if ( p->refc > 0 ) {
            break ;
        } else {
            struct ListRC* del = p ;
            p = p->next ;
            printf( "%d " , del->data ) ;
            free( del ) ;
        }
    }
    printf( "\n" ) ;
}

int main() {
    struct ListRC* a = cons( 3 , cons( 4 , NULL ) ) ;
    struct ListRC* b = cons( 1 , copy( a->next ) ) ;
    struct ListRC* c = cons( 2 , copy( a ) ) ;
    /* (A) */
    // [結果解答欄] (数値が表示されない場所もある)

    free_list( a ) ; // -----
    free_list( b ) ; // -----
    free_list( c ) ; // -----
    return 0 ;
}

| [イメージ図]
```

4 説明問題 (25)

以下の3つの説明問題から2つを選んで答えよ。(13+12)

- ハッシュ法におけるオープンアドレス法とチェイン法の違いについて、イメージ図などを交えながら説明せよ。
- ガベージコレクタとその処理の内容について説明せよ。
- 参照カウンタ法がOSの機能の中で使われている事例を説明せよ。

5 データベースの設計 (25)

以下に示すような条件で、このデータ構造をハッシュ法で管理したい。

とある地区の全住民の「名前と生年月日」のデータで、データ件数は約 1000 件とする。名前をキーとして生年月日を検索したい。この処理にふさわしい (1) C 言語でのデータ構造と (2) ハッシュ表の宣言、および (3) ハッシュ関数について具体的にソースコードで記載し、(4) データを検索する処理について説明せよ。(4) については言葉での説明で良い。

ただし、処理時間のオーダーは $O(1)$ であること。(8+5+7+5)