

5つの設問の中から、4つを選んで回答せよ。

1 動作トレース (25)

配列の要素に同じものが複数含まれているか判定するための `rec_uniq()` について、以下のプログラムの実行結果を答えよ。

```
#include <stdio.h>

int a[ 4 ] = { 111 , 222 , 333 , 222 } ;
int b[ 3 ] = { 11 , 22 , 33 , 44 } ;

int rec_uniq( int array[] , int size ) {
    if ( size <= 1 ) {
        return 1 ;
    } else {
        int tail = array[ size - 1 ] ;
        for( int i = 0 ; i < size - 1 ; i++ ) {
            printf( "%d %d\n" , array[ i ] , tail ) ;
            if ( array[ i ] == tail )
                return 0 ;
        }
        return rec_uniq( array , size - 1 ) ;
    }
}

void main() {
    if ( rec_uniq( a , 4 ) )
        printf( "Uniq\n" ) ;
    if ( rec_uniq( b , 4 ) )
        printf( "Uniq\n" ) ;
}

[回答欄]
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
```

2 処理時間の見積もり (25)

- データ件数 N に対する処理時間が $T(N) = T_\alpha + T_\beta \times 1.1^N + T_\gamma \times N \log N$ で示される場合、オーダー記法で表せ。(5)
- データの並び替えを、最大選択法とクイックソートで行う場合、データ件数 10 件のとき、最大選択法のプログラムは 2msec、クイックソートでは 10msec であった。
データ件数 100 件の時、最大選択法とクイックソートではどちらが速いか、具体的な数字を交えて答えよ。(10)
- 前設問 1 の `rec_uniq()` において、データ件数を N としたとき、同じ値がない配列で呼び出した場合の処理時間を、オーダー記法で答えよ。(10)

3 穴埋め問題 (25)

```
struct NameUrlCount {      | 以下のプログラムで、
char name[ 10 ] ;         | 利用者が URL を何回ページを見た回数の情報を、
char* url ;               | 読み込んでその結果を表示したい。
int count ;               | プログラムの下線部にふさわしい処理を答えよ。
};
void main() {              | (入力例)
struct NameUrlCount* table ; | 3
int size ;                | tsaitoh http://tsaitoh.net/=t-saitoh/ctf 10
scanf( "%d" ,             | tomoko https://www.google.com/ 123
        ,                 | ayuka https://www.pixiv.net/ 42
        ~~~~~(A) );
table = (struct NameUrlCount*)malloc( ~~~~~(B) * size ) ;
if ( table != ~~~~~(C) ) {
    // データの入力
    for( int i = 0 ; i < size ; i++ ) {
        char buff[ 1024 ] ;
        if ( scanf( "%s %s %d" ,
                    table[i].name , buff , ~~~~~(D) ) != 3 )
            break ;
        if ( (table[i].url = ~~~~~(E) ) == NULL )
            break ;
        strcpy( ~~~~~(F) , buff ) ;
    }
    // 覚えたデータの出力と url の文字列を廃棄
    for( int i = 0 ; i < size ; i++ ) {
        printf( "name=%s url=%s count=%d\n" ,
                table[i].name , table[i].url , table[i].count ) ;
        free( ~~~~~(G) ) ;
    }
    ~~~~~(H)
}
}
```

4 説明 (25)

1. プログラムを作成するにあたり、メモリを膨大に消費するプログラムがあった場合、どのような問題が発生することが想定されるか、具体例を交えながら説明せよ。(10)
2. 以下のプログラム例を使って、(1),(2),(3) の用語の中から 2 つについて、説明せよ。(7+8)

```
void foo( int x ) {          | void bar( int* px ) {
    static int y = 123 ;    |     (*px)++ ;
    x++ , y++ ;            |     printf( "%d\n" , *px ) ;
    printf( "%d %d\n" , x , y ) ; | }
}
void main() {               | 用語
    int x = 12 ;           | (1) 値渡し
    foo( x ) ;             | (2) 変数の寿命
    bar( &x ) ;            | (3) 静的局所変数
}
}
```

5 プログラム作成問題 (25)

モビルスーツのパイロットの ID, 名前, 年齢の情報が (id,name,age) の要素名で配列 `pilot[]` で与えられる。また、モビルスーツの名前, モデル番号, パイロットの ID が (name,model,pid) の要素名で配列 `ms[]` で与えられる。

(1) この配列 `pilot[], ms[]` のための構造体 `Pilot, MobileSuit` の宣言を示し, (2) 配列 `pilot[]` に記載された全てのパイロットの名前, モデル番号を表示し、そのパイロットの中で最も若い人の年齢を出力するプログラムを示せ。(注意: Miorine のように `ms[]` に含まれない場合は、モデル番号の代わりに”-”を出力すること)

```
struct Pilot {                                | struct MobileSuit { // (1) 構造体の宣言 (8)
|
|
|
|
|
|
|
|
};                                           | };
|
|
|
|
|
|
|
|
struct Pilot pilot[] = {                    | struct MobileSuit ms[] = {
  // id , name , age                         | // name , model , pid
  { 1001 , "Amuro" , 15 } ,                  | { "AERIAL" , "XVX-016" , 9001 } ,
  { 2002 , "Quattro" , 27 } ,              | { "DILANZA" , "MD-0032G" , 9003 } ,
  { 9001 , "Suletta" , 17 } ,              | { "GUNDAM" , "RX-78" , 1001 } ,
  { 9002 , "Miorine" , 16 } ,              | { "Type100" , "MSN-00100" , 2002 } ,
  { 9003 , "Guel" , 18 } ,                 | };
};
```

// (2) パイロット名とモデル番号を表示し (9)、パイロットの最低年齢を求める処理 (8)