

以下の 6 つの設問の中から、5 つを選んで解答せよ。

1 字句解析、構文解析 (20)

((設問 1))

以下の正規表現は、(A)～(E) の文字列にマッチするか、それぞれ○×で答えよ。

`[a-z]+_[a-z0-9]*`

☐ 括弧内に含まれる 1 文字にマッチする。

`a-b` は文字の範囲を表す

`+` 1 個以上の繰返し

`*` 0 個以上の繰返し

(A) `a` (B) `int_` (C) `a_1_2` (D) `g00gle` (E) `_12cd`

((設問 2))

数値 `<number>` を以下の BNF 記法で定義する。

(A)～(E) が`<number>`として扱われるかを、それぞれ○×で答えよ。

`<number> ::= <digits> | <digits>E<digits> | <digits>E<sign><digits>`

`<digits> ::= <digit> | <digits><digit>`

`<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`

`<sign> ::= + | -`

(A) `12E` (B) `-123` (C) `12E-3` (D) `+34E-56` (E) `12+E10`

2 基礎問題と再帰呼出し

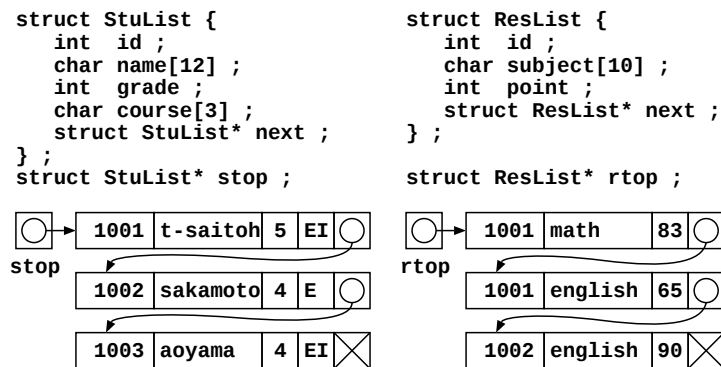
以下のプログラムの実行結果を答えよ。

```
void print( int a[] , int idx , int max ) { | int b[3] = { 3, 2, 1 } ;
    if ( idx < max ) { | int c[5] = {
        printf( "(" ) ; | 50, 20, 70, 10, 40 } ;
        print( a , idx * 2 + 1 , max ) ; |
        printf( "-%d-" , a[ idx ] ) ; | void main() {
        print( a , idx * 2 + 2 , max ) ; |     print( b , 0 , 3 ) ;
        printf( ")" ) ; |     printf( "\n" ) ;
    } |     print( c , 0 , 5 ) ;
} | }
```

3 単純リストで組み合わせ処理 (20)

2つの表形式のデータが以下の図のように保存されている。このデータの中から、成績が80点以上の「科目名, 学生名, 点数」を対応関係が判るように表示するプログラムを作成せよ。
ResultList の ID と StuList の ID は、それぞれ対応している。

((出力例))
math t-saitoh 83
english sakamoto 90



4 説明問題 (20)

- データ部が整数の、番兵付きの双方向循環リストについて、(a) そのデータ構造の宣言を示し、(b) イメージ図を交えてデータ構造を説明せよ。
- 2分探索木において、左右が不均等な木はどういう場合に生成されるのか具体例を交えながら説明せよ。

5 2分木の穴埋め問題 (20)

```

struct STree {
    char    str[ 10 ] ;
    struct STree* left ;
    struct STree* right ;
} ;

struct STree* top = NULL ;

void main() {
    add( "def" ) ;
    add( "abc" ) ;
    add( "hij" ) ;
}

void add( char s[] ) {
    struct STree* tcons( char s[] ,
                          struct STree* l , struct STree* r )
    {
        struct STree* ans ;

        ans = ~~~~~(A)3

        if ( ans != NULL ) {
            strcpy( ans->str , s ) ;
            ~~~~~(h)
            ans->left = l ;
            ans->right = r ;
        } ~~~~~(i)
        return ans ;
    }

    ~~~~~(B)3 ~~~~~(C)2
    while( *tail != NULL ) {
        int c = strcmp( (*tail)->str , s ) ;
        ~~~~~(j) ~~~~~(k)

        if ( ~~~~~ )
            ~~~~~(D)3
            break ;
        else if ( c > 0 )
            ~~~~~(l)

            tail = ~~~~~(E)3
        else
            tail = &( (*tail)->right ) ;
            ~~~~~(m)
    }
    if ( ~~~~~ == NULL )
        ~~~~~(F)3
    *tail = ~~~~~(G)3
}

```

// 設問1
// プログラム中の下線部 (A)-(G)
// に相応しい処理を埋めよ。

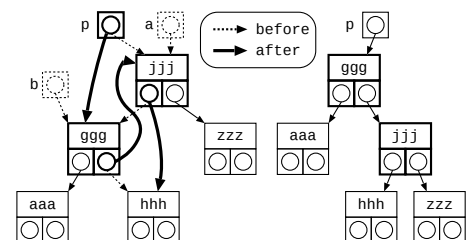
```

//      top
//      / \
//     def
//    /   \
//   abc   hij

```

6 型と AVL 木 (20)

1. 前設問の下線部 (h)-(m) の型を答えよ。 (2x6=12)
2. 前設問の STree の 2 分木で AVL 木の右回転処理を行うプログラムを、以下のような関数で作成したい。処理を完成させよ。



```

(h) | struct STree* avl_rot_right( struct STree* p ) {
      |     struct STree* a = p ;
      |     struct STree* b = p->left ;
      |
      |     ~~~~~(A)3
      |     ~~~~~(B)3
      |     p = ~~~~~(C)2
      |     return p ;
      | }

```