

以下の5つの設問の中から、4つを選んで解答せよ。

1 基本問題 (25)

| | |
|--|---|
| <pre> struct Tree { int data ; struct Tree* left ; struct Tree* right ; } ; struct Tree* top = NULL ; void entry_tree(int value) { struct Tree** tail = &top ; while(*tail != NULL) { if ((*tail)->data == value) break ; else if ((*tail)->data > value) tail = &((*tail)->left) ; else tail = &((*tail)->right) ; } if (*tail == NULL) { *tail = (struct Tree*)malloc(sizeof(struct Tree)) ; if (*tail != NULL) { (*tail)->data = value ; (*tail)->left = (*tail)->right = NULL ; } } } void print_tree(struct Tree* p) { if (p != NULL) { print_tree() ; printf("%d\n" ,) ; } } </pre> | <p>授業の中で示した2分探索木のプログラムについて、 (1) 下線部 (A)-(E) の型を答えよ。 (2) 全データを昇順で表示する print_tree() と、指定した値を探す find_tree() の下線部 (F)-(K) にふさわしい処理を答えよ</p> |
| <pre> (A) _____ (B) _____ (C) _____ (D) _____ (E) _____ </pre> <p style="text-align: right;">(2x5)</p> | <pre> int find_tree(struct Tree* p , int key) { while(_____) { if (_____) return 1 ; // find! else if (p->data > key) p = p->left ; else ; } return 0 ; } </pre> <p style="text-align: right;">(2x3+3x3)</p> |

2 正規表現とコンパイラ (25)

1. 正規表現 $[0-9][0-9A-F]^*$ が表現する文字列の集合の要素となるものはどれか。要素となるものに○、ならないものに×を記入せよ。

(1) 2-A ____ (2) FA9 ____ (3) 1AC ____ (4) 3B* ____ (4x4)

2. 以下の説明の下線部にふさわしい単語を答えよ。

コンパイラの内部では、演算子や数値や変数名といったトークンに切り分ける (A)_____ 解析処理、文法にあわせた (B)_____ 木を生成する (B) 解析処理が行われる。この後、効率の良い機械語にするための (C)_____ 化を行ってから、機械語生成処理を行う。(3x3)

3 双方向リスト (25)

双方向リストの途中にデータを挿入する処理、削除する処理のプログラムで、(A)~(E) に適切な処理を答えよ。(5x5)

```

struct StrDeque {
    char str[ 20 ];
    struct StrDeque* prev ;
    struct StrDeque* next ;
} ;

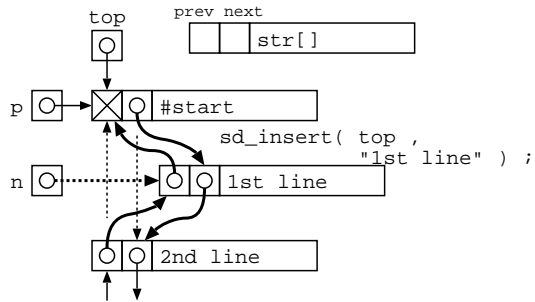
void main() {
    struct StrDeque* top =
        sd_cons( "#start",NULL ,
                sd_cons( "2nd line",NULL ,
                          sd_cons( "#end",NULL,NULL ) ) ) ;
    top->next->prev = top ;
    top->next->next->prev = top->next ;
    sd_insert( top , "1st line" ) ;
    sd_delete( top->next ) ;
}

```

```

struct StrDeque* sd_cons( char* s ,
                          struct StrDeque *p ,
                          struct StrDeque *n ) {
    struct StrDeque* nw =
        (struct StrDeque*)malloc(
            sizeof(struct StrDeque))
    if ( nw != NULL ) {
        strcpy( nw->str , s ) ;
        nw->prev = p ;
        nw->next = n ;
    }
    return nw ;
}

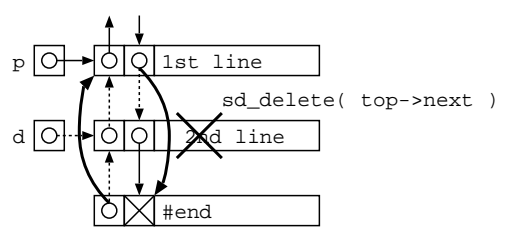
```



```

void sd_insert( struct StrDeque* p , char* s ) {
    struct StrDeque* n = sd_cons( s , p , p->next ) ;
    _____(A)
    _____(B)
}

```



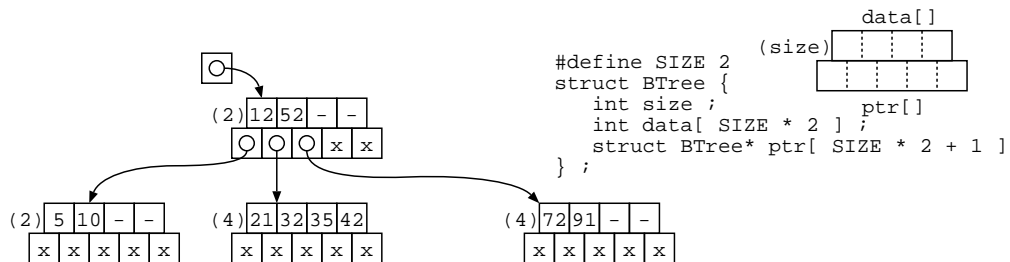
```

void sd_delete( struct StrDeque* p ) {
    struct StrDeque* d = p->next ;
    _____(C)
    _____(D)
    _____(E)
}

```

4 説明問題 (12+13)

1. データが N 件、保存されている 2 分探索木について、データの検索時間が最短の場合と最悪の場合について、図と探索時間のオーダ記法などを交えながら説明せよ
2. B 木に要素を追加する時に、木の構造が偏らないようにするために、どのような処理が行われるか、下図に 40 を追加する処理を例に説明せよ。



5 2分木と状態遷移図 (25)

```

struct SMTree* smt_cons( char* stn ,
                        struct SMTree* alp ,
                        struct SMTree* num )
{
    struct SMTree* ans =
        (struct SMTree*)malloc(sizeof(struct SMTree)) ;
    if ( ans != NULL ) {
        ans->state = stn ;
        ans->alpha = alp ;
        ans->digit = num ;
    }
    return ans ;
}

```

```

void check( SMTree* p , char* str ) {
    for( int i = 0 ; p != NULL && str[i] != '\0' ; i++ ) {
        printf( "%s:%c\n" , p->state , str[i] ) ;
        if ( isalpha( str[i] ) ) // 英字 (a-z)
            p = p->alpha ;
        else if ( isdigit( str[i] ) ) // 数字 (0-9)
            p = p->digit ;
        else
            p = NULL ;
    }
    if ( p != NULL )
        printf( "Valid\n" ) ;
    else
        printf( "Error\n" ) ;
}

```

[output]

```

-----
-----
-----
-----
-----
-----
-----
-----

```

```

int main() {
    struct SMTree* st =
        smt_cons( "ST" ,
                 smt_cons( "A" ,
                           NULL ,
                           smt_cons( "AN",NULL,NULL ) ) ,
                 NULL ) ;
    struct SMTree* a = st->alpha ;
    struct SMTree* an = st->alpha->digit ;
    a->alpha = a ;
    an->alpha = an ;
    an->digit = an ;
    check( st , "ab3" ) ;
    check( st , "12" ) ;
    return 0 ;
}

```

(15+10)

関数の説明

isalpha(c) - 英字 (A-Z,a-z)

isdigit(c) - 数字 (0-9)

左に示すプログラムの
実行結果を答えよ。

[output] 欄に記述すること。

また、st のデータ

構造のイメージ図 (下図)

の未記入欄を完成させよ。

