情報構造論　後期中間　　番号：

4EI　　(1/3)　　氏名：

★の付いた６つの設問から、５つを選択し回答せよ。

それ以上の回答がされている場合は、1/3 にて加算する。設問中 #include<...> は、省略する。

## 1　単純線形リストの基礎確認

```
struct List {
    int          data ;
    struct List* next ;
} ;
struct List* cons( int x , struct List* n )
{   struct List* ans = (struct List*)malloc( sizeof( struct List ) ) ;
    if ( ans != NULL ) {
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
}
struct List* find( struct List* p , int key )
{   /* 見付けたら、そのリスト。無かったら、 NULL を返す */
    if ( p == NULL )
         ~~~~~~~~~(A)
        return p ;     /* 見付からなかった */
        ~~~~~~~~(B)
    else if ( p->data == key )
              ~~~~~~~~~~~~~~(C)
        return p ;     /* 見付けた */
        ~~~~~~~~(D)
    else
        return find( p->next , key ) ; /* 残りから探す */
}       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~(E)
void main()
{
    struct List* list = cons( 11 , cons( 22 , cons( 33 , NULL ) ) ) ;
                ~~~~(F)
    find( list , 33 ) ;
}   ~~~~~~~~~~~~~~~~(G)
```

- (F) に代入されたリストのイメージ図を示せ。
- (G) の処理を実行した場合の、処理順序の流れを
  (A) ～ (E) の記号を用いて示せ。

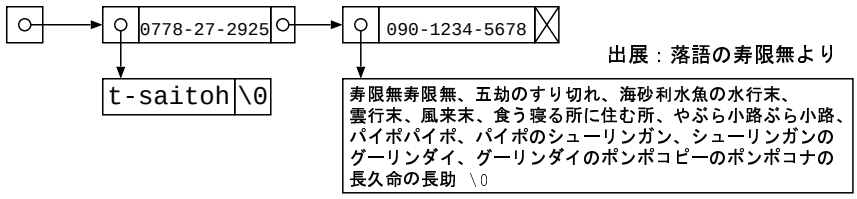
## 2　文字列を交えたデータ構造

文字列を含むリスト構造を扱う以下のプログラムを作成した。しかし下線部 (A),(B) の間違いのため、動作できない。それぞれの間違いの理由を説明し、適切な命令に書き換えよ。

ヒント：strcpy(),strdup()

```
struct NamePhoneList {
    char*  name ;         /* とても長い名前の人もいるかも */
    char   phone[ 16 ] ; /* 電話番号は長くても 12 桁だろう */
    struct NamePhoneList* next ;
} ;
void main()
{   char  t_name[ 1000 ] , t_phone[ 100 ] ;
    struct NamePhoneList* top = NULL ;

    while( scanf( "%s %s" , t_name , t_phone ) == 2 ) {
        struct NamePhoneList* n ;
        n = (struct NamePhoneList*)malloc(
                sizeof( struct NamePhoneList ) ) ;
        if ( n != NULL ) {
            n->name  = t_name ;
            ~~~~~~~~~~~~~~~~~~(A)
            n->phone = t_phone ;
            ~~~~~~~~~~~~~~~~~~~(B)
            n->next = top ;  /* 先頭にリストを挿入 */
            top = n ;
        }
    }
    for( ; top != NULL ; top = top->next ) { /* 全表示 */
        printf( "%s %s\n" , top->name , top->phone ) ;
    }
}
```



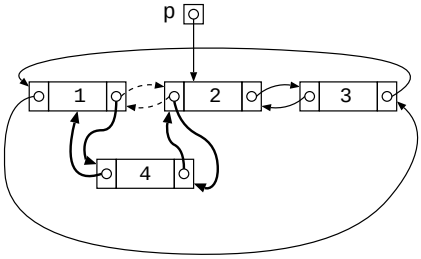

**(A)**

**(B)**

## 3　双方向リングリストのプログラム

次のようなデータ構造にて双方向のリングリストを作成したい。そして、この様な双方向リングリストにて、目的の場所の前に 1 件のリスト構造を挿入する関数 bdlist_insert() を作成したい。下図を参考に関数内部を作成せよ。

```
struct BDList {
    int             data ;
    struct BDList* prev ;
    struct BDList* next ;
} ;
```




```
/* ポインタ p の前に key を追加したい */
void bdlist_insert( struct BDList* p , int key )
{
    /* この部分を作成せよ */
}
```

## 4　２分木のプログラム

```
struct Tree {
    int          data ;
    struct Tree* left ;
    struct Tree* right ;
} ;
struct Tree* top = NULL ;

void add_tree( int x )
{
    struct Tree** ppt = &top ;
                        ~~~~(A)
    while( *ppt != NULL ) {
        if ( (*ppt) == x ) {
            break ; /* 同じデータならループを脱出 */
        } else if ( (*ppt)->data > x ) {
                    ~~~~~~~~~~~~(B)
            ppt = &( (*ppt)->left ) ;  /* 左の枝 */
                    ~~~~~~~~~~~~(C)
        } else {
            ppt = &( (*ppt)->right ) ; /* 右の枝 */
        }
    }
    if ( *ppt == NULL ) { /* ヒント：前の break に注目 */
         ~~~~~~~~~~~~(D)
        *ppt = (struct Tree*)malloc( sizeof( struct Tree ) ) ;
        if ( *ppt != NULL ) {
             ~~~~~~~~~~~~(E)
            (*ppt)->data  = x ;
            (*ppt)->left  = NULL ;
            (*ppt)->right = NULL ;
        }
    }
}
```

- プログラム中の (A) ～ (C) の型を述べよ。

  **(A)**

  **(B)**

  **(C)**

- (D) と (E) において、それぞれ NULL と比較している理由を述べよ。

  **(D)**

  **(E)**

## 5　プログラム作成問題

前述の問 4の２分木において、指定された数より等しいか大きい (つまり以上の) データの件数を数えるプログラムを作成したい。以下のプログラムの内部を作成せよ。

```
/* key の値より大きい (or 等しい) データの件数を数える */
int count_ge( struct Tree* p , int key )
{
    /* この部分を作成する */
}
```

## 6　説明

1. 設問 1における関数`find()` の様なプログラムスタイルはどの様に呼ばれるか。および、利点や欠点や特徴を交えて、その具体的に説明を加えよ。

2. 設問 4における２分木データ構造`struct Tree` について、説明せよ。データ構造の要素の意味や、その方式による処理速度面での特徴を、具体的に述べること。