情報構造論　前期期末試験 (出題：斉藤)　番号：
4EI　2004/09/15,(1/2)　氏名：

1　リスト処理基本

複素数を要素とするリストを扱うプログラムを作りたい。以下の設問に答えよ。

```
struct ComplexList {
    float  re , im ;           /* 実部と虚部      */
    struct ComplexList* next ; /* 次へのポインタ */
} ;
struct ComplexList*  top  = NULL ;
struct ComplexList** tail = &top ;
                            ~~~~A
void add( float r , float i )
{   struct ComplexList* n ;
    n = (struct ComplexList*)malloc( sizeof( struct ComplexList ) ) ;
    if ( n != NULL ) {                ~~~~~~~~~~~~~~~~~~~~~~~~~~~B
         ~~~~~~~~C
        n->re = r ;  n->im = i ;
                           ~~~D
        n->next = NULL ;
                  ~~~~E
        *tail = n ;
        tail = &( (*tail) ->next) ;
    }             ~~~~~~~F
}
float distance( struct ComplexList* p , struct ComplexList*q )
{
    float  dre = p->re - q->re ; /* 設問２で穴埋め */

    float  dim =                            ;
                ~~~~~~~~~~~~~~~~~~~~~~~~~~G
    return sqrt(                             ) ; /* sqrt= 平方根 */
}               ~~~~~~~~~~~~~~~~~~~~~~~~~~H
void print( struct ComplexList* p )
{   for( ; p != NULL ; p = p->next )
    printf( "%f + j%f\n" , p->re , p->im ) ;
}
void main()
{   float  max = 0 ;
    add( 1.0 , 1.5 ) ;  /* リスト生成 (1) */
    add( -2.4 , 1.2 ) ; /* リスト生成 (2) */
    add( 4.0 , 6.0 ) ;  /* リスト生成 (3) */
    print( top ) ;

    /* 設問２で最大の２点間距離を探す処理を書く場所 */
}
```

1.1　リスト構造の理解確認 ($\times 10$)

プログラム中のリスト生成 (3) が終了した時点で、出来上がったリストの状態を図で説明せよ。ただし`top,tail` の指し示す場所も図に記入すること。

1.2　説明問題 ($5 \times 2{=}10$)

1. 下線部 (C) の NULL 比較を行う理由を答えよ。

2. 下線部 (E) の NULL 代入を行う理由を答えよ。

2　プログラム作成 ($\times$ 20)

設問 1 で与えられたリストの要素の複素数で、２つの複素数の座標の距離が最も離れた最大距離を計算するプログラムを、作りたい。２つの複素数間の距離を求める関数 `distance()` の下線部 G , H を埋めた後、`main()` の最後の部分に書くプログラムを作成せよ。なお、必要に応じて変数は宣言すれば良い。

情報構造論　前期期末試験 (出題：斉藤)　番号：
4EI　2004/09/15,(2/2)　氏名：

3　設問 1 の下線部 A,B,D,F の型を答えよ (5 × 4=20)

A

B

D

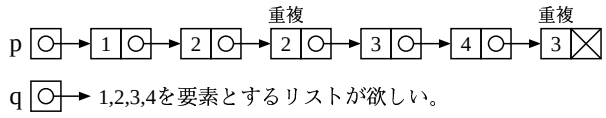
F

4　説明問題 (× 20)

リスト構造ついて、３つのポイントをあげて、利点と欠点を具体的に説明せよ。ただし、３つのうちの１つは、処理速度についてオーダ記法などを交えた説明をすること。

5　リストによる集合 (× 20)

リスト構造によって集合を表現したい。しかし、与えられたリストに同じ値の要素が重複している場合があるとする。この重複をなくしたリストを新しく生成したい。以下のプログラム中の`no_duplicate` の内部を作成せよ。必要に応じて `cons` を利用すれば良い。ただし、集合として扱うため要素の順序は問わない。




```
struct List {
   int          data ;
   struct List* next ;
} ;

struct List* cons( int x , struct List* p )
{  struct List* n = (struct List*)malloc( sizeof( struct List ) ) ;
   if ( n != NULL ) {
      n->data = x ;  n->next = p ;
   }
   return n ;
}
void print( struct List* p )
{  for( ; p != NULL ; p = p->next )
      printf( "%d " , p->data ) ;
}
struct List* no_duplicate( struct List* p )
{
   /* この関数の内部を作成せよ */
}
void main() {
   struct List* lst =
          cons(1,cons(2,cons(2,cons(3,cons(4,cons(3,NULL))))));
   print( no_duplicate( lst ) ) ; /* 重複を除いたリストを表示 */
}
```