

1 レポート課題相当と型

```
struct NamePhoneList {          | void np_print( struct NamePhoneList* p ) {
  char name[ 10 ] ;             |   for( ; p != NULL ; p = p->next )
  int  phone ;                  |   printf( "%s %d\n" , p->name , p->phone ) ;
  struct NamePhoneList* next ;  | }                                     ~~~~~(A)
} ;                               | -----
                                | [設問 1]
                                | このプログラムの (G) の終了時点
                                | で出来上がったデータ構造の
                                | イメージ図を描け。(10)
struct NamePhoneList* top = NULL ;
struct NamePhoneList** tail = &top ;

struct NamePhoneList* npcons( char*s , int p ,
                              struct NamePhoneList* n ) {
  struct NamePhoneList* ans ;
  if ( (ans = (struct NamePhoneList*)malloc(
    sizeof( struct NamePhoneList ) ) ) != NULL ) {
    strcpy( ans->name , s ) ;
    ~~~(B)
    ans->phone = p ;
    ans->next = n ;
  }
  return ans ;
}

void main() {
  *tail = npcons( "t-saitoh" , 271234 , NULL ) ;
  ~~~~~(C)
  tail = &((*tail)->next) ;
  ~~~~~(D)
  top = npcons( "mitsuki" , 621111 , top ) ;
  *tail = npcons( "ayuka" , 111111 , NULL ) ;
  tail = &( (*tail) -> next ) ; /* (G) */
  ~~~~~(F)
}
```

[設問 2]

下線部 (A) ~ (F) の型を答えよ。(2x5)

(A) _____ (B) _____ (C) _____
(D) _____ (E) _____

2 リストの廃棄処理

前設問の NamePhoneList のプログラムでは、処理終了後のリスト廃棄処理が記述されていない。
(G) の次の行に、以下のような処理を呼び出すとして、関数 np_freelist() を記述せよ。

```
/* (G) の次の行 */
np_freelist( top ) ;

/* 以下の関数を記述せよ */
void np_freelist(
```

3 集合問題

```
struct List {
    int data ;
    struct List* next ;
};
void set_print( struct List* p ) {
    for( ; p != NULL ; p = p->next )
        printf( "%d " , p->data ) ;
    printf( "\n" ) ;
}
struct List* cons( int x , struct List* n ) {
    struct List* ans ;
    ans = ~~~~~(A)3
    if ( ans != NULL ) {
        ans->data = ~~~~~(B)2
    } ~~~~~(C)3
    return ans ;
}
struct List* set_and( struct List*a , struct List*b ) {
    struct List* ans = NULL ;
    struct List* p ; struct List* q ;
    for( p = a ; ~~~~~ ; ~~~~~ ) {
        for( q = b ; ~~~~~ ; ~~~~~ ) {
            if ( p->data ~~~~~ )
                break ; ~~~~~(F)3
            if ( p == NULL )
                ~~~~~(G)3
        }
    }
    return ans ;
}
void main() {
    struct List* a = cons(1,cons(2,cons(3,NULL))) ;
    struct List* b = cons(2,cons(3,cons(4,NULL))) ;
    set_print( set_and( a , b ) ) ;
}
```

【設問】
積集合を求めるプログラムを作成した。
適切に動くように、プログラム中の
(A)～(G)に相応しい処理を、答えよ。
集合なのでリスト内の順序は問わない。

4 説明問題

1. 双方向循環リストについて、データ構造の宣言とそのイメージ図を記述し説明せよ。(10)
2. 配列を用いて待ち行列を作る場合のリングバッファについて図を交えながら説明せよ。(10)

5 応用・リストの再帰処理

```
int rec_find( struct List*p , int k ) { | [設問]
  if ( p == NULL ) { | 再帰処理を用いてリスト処理のプログラム
    return 0 ; | を記述した。以下のプログラムを実行した
  } else { | 時に表示される内容を答えよ。
    printf( "%d:%d\n" , p->data , k ) ; | (20)
    ~~~~~
    if ( p->data == k ) |
      return 1 ; |
    else |
      return rec_find( p->next , k ) ; |
    } |
  } |
} |

void main() {
  struct List* a = cons(1,cons(2,cons(3,NULL))) ;
  struct List* b = cons(2,cons(3,NULL)) ;

  struct List* c = rec_and( a , b , NULL ) ;
}

struct List* rec_and( struct List* a , struct List* b , struct List* ans ) {
  if ( a == NULL )
    return ans ;
  else if ( rec_find( b , a->data ) )
    return rec_and( a->next , b , cons( a->data , ans ) ) ;
  else
    return rec_and( a->next , b , ans ) ;
}
```

6 記述問題

名前と数学, 理科, 英語の成績のデータを、右図に示すように保存し、その処理プログラムを作る。
このリスト構造に相応しいデータ構造を宣言し、このデータで3教科の合計点が最も大きい人の名前を出力するプログラムを作成せよ。
図のデータ構造を生成する処理は、記載しなくて良い。

