

以下の5つ設問から4つを選んで回答せよ。

1 基本動作トレース (25)

```

struct List* cons( int x , struct List* n ) {
    struct List* ans ;
    ans = (struct List*)malloc( sizeof( struct List ) ) ;
    if ( ans != NULL ) {
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
}

int main() {
    struct List* lst1 = cons( 11 , cons( 22 , NULL ) ) ;
    struct List* lst2 = cons( 22 , cons( 33 , NULL ) ) ;
    struct List* p ;
    struct List* q ;
    for( p = lst1 ; p != NULL ; p = p->next )
        ~~~~~(A) ~~~~~(B) ~~~~~(C)
        for( q = lst2 ; q != NULL ; q = q->next )
            ~~~~~(D) ~~~~~(E) ~~~~~(F)
            if ( p->data == q->data )
                ~~~~~(G)
                break ;
                ~~~~~(H)
    return 0 ;
}

```

```

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

```

(1) このプログラムの (A)-(G) が止まるまでの実行順序を答えよ。(15)

(2) プログラムが "return 0" 時点で lst1, lst2, p, q の指し示すデータのイメージ図を記載せよ。
(H) の直後に注意 (10)

2 説明問題 (25)

1. 前設問の補助関数 `cons()` の内部で、`ans != NULL` の比較を行う理由を説明せよ。(6)
2. 「リスト構造」と「C言語の基本的な配列」を比べ、リスト構造が優れている点を2つ、(6x2)
3. 逆に、「C言語の基本的な配列」の方が優れている点を1つ、説明せよ。(7)

3 プログラム穴埋め問題 (25)

以下のプログラムは、データを昇順でリストに保存するプログラムで、処理を単純にするため、データの先頭と末尾に 0,999 を入れるものとする。

ヒント : `asc_ins()` の `pp` は、挿入する 1 つ前のリストを指し示すための変数とする。

以下の設問に答えよ。

```
void print( struct List* p ) {  
    for( ; p != NULL ; p = p->next )  
        ~~~~~(A)  
        printf( "%d " , p->data ) ;  
        ~~~~~(B)  
    printf( "\n" ) ;  
}  
  
void asc_ins( struct List* p , int x ) {  
    struct List* pp = p ;  
  
    for( ; p != NULL ; ~~~~~ ) {  
        ~~~~~(C)3  
        if ( ~~~~~ )  
            break ;  
        pp = p ;  
    } ~~~(E)  
  
    ~~~~~ = cons( ~~~~~ , ~~~~~ ) ;  
    ~~~~~(F)4  
  
int main() {  
    struct List* top = cons( 0 , cons( 999 , NULL ) ) ;  
    ~~~~~(G)  
    asc_ins( top , 80 ) ; print( top ) ;    // 出力  
    ~~~~~(H)    // 0 80 999  
    asc_ins( top , 25 ) ; print( top ) ;    // 0 25 80 999  
    asc_ins( top , 50 ) ; print( top ) ;    // 0 25 50 80 999  
    return 0 ;  
}
```

(1) 下線部の型を答えよ (2x5)

(A) ~~~~~
(B) ~~~~~
(E) ~~~~~
(G) ~~~~~
(H) ~~~~~

(2) (C), (D), (F) に、(3+3+4) 出力に示した結果となるように処理を記入せよ。

(3) `top` に保存されるデータ件数を `N` としたとき、`asc_ins()` の処理時間をオーダー記法で答えよ。(5)

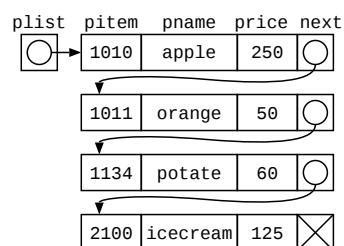
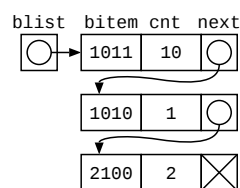
4 説明問題 (25)

- 配列を用いた待ち行列を作る場合の方法を、データの出し入れする際の処理の内容や図を交えて説明せよ。(13)
- リスト構造を用いたスタックについて、データの出し入れする際の具体的なプログラムもしくは図を交えて、説明せよ。(12)

5 基本課題問題

```
struct BuyList {  
    int bitem ;  
    int cnt ;  
    struct BuyList* next ;  
} ;  
struct BuyList* blist ;
```

[PriceList]



output
1011 orange 50*10 500
1010 apple 250*1 250
2100 icecream 125*2 250
total 1000

上図のように、購入リスト BuyList, 単価情報リスト PriceList が与えられた場合、その購入結果を、図中 output のように出力するプログラムを作成したい。

図中の bitem は、購入する品物につけられた固有の ID で、単価情報にはそれに対応した pitem という項目が存在する。

1. このプログラムの PriceList のデータ構造にふさわしいリスト構造の宣言を答えよ。
2. 図のデータ構造を生成する処理は、別に作られているものとする。

この時、購入リスト blist が与えられたら、output のような出力を行う関数、
`print_buy_list( struct BuyList* blist )` を作成せよ。