

6つの設問の中から5つを選んで回答せよ。

全問回答の場合には、各設問を採点した後、最低点の設問は加点しない。

1 基本的なリスト操作

リスト構造を用いた簡単なプログラムを作った。

実行結果と、その処理で生成されるデータ構造のイメージ図を答えよ。

```

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

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

void main() {
    struct List* top = cons( 12 , cons( 23 , cons( 34 , NULL ) ) ) ;
                                表示内容          イメージ図

    print( top ) ;
    ~~~~~(A)          (A)-----
    print( even( top ) ) ;
} ~~~~~(B) (B)-----
    
```

2 ちょっと応用問題

同じく前述のデータ構造を使い以下のようなプログラムを作った。下線部の型や表示内容を回答せよ。

```

struct List* rp = cons( -1 , NULL ) ;
                                ~~~~~(A)
struct List** wp = &rp->next ;

void put( int x ) {
    *wp = cons( x , NULL ) ;
    ~~~(B)    ~~~(C)
    wp = &((*wp)->next) ;
} ~~~~~(D)

int get() {
    int ans = rp->data ;
    ~~~~~(E)
    struct List* del = rp ;

    rp = rp->next ;
    ~~~~~(F)
    free( del ) ;

    return ans ;
} ~~~(G)

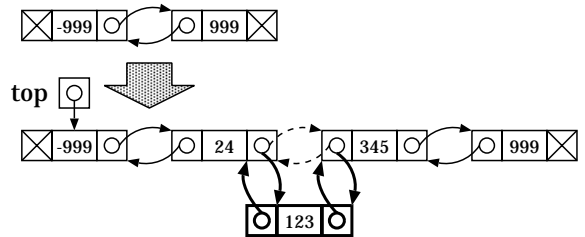
void main() {
    put( 123 ) ;
    put( 234 ) ;
    put( 345 ) ;
    printf( "%d\n" , get() ) ;
    ~~~~~(H)
    printf( "%d\n" , get() ) ;
    ~~~~~(I)
    printf( "%d\n" , get() ) ;
} ~~~~~(J)

    回答欄 (A) ~ (G) は型 , (H) ~ (J) は表示内容
    (A)----- (F)-----
    (B)----- (G)-----
    (C)----- (H)-----
    (D)----- (I)-----
    (E)----- (J)-----
    
```

3 プログラム穴埋め

整数データを記憶する双方向リストにおいて、データを昇順になるようにデータを挿入するinsert(値)を作成せよ。ただしリストは初期状態でデータ先頭に-999, 末尾に999を入れてあり、取り扱うデータはこの値の範囲内とする。

```
struct BDList {
    struct BDList* prev ;
    int data ;
    struct BDList* next ;
};
struct BDList* top = ... 初期化... ;
```



```
void insert( int y ) {
    struct BDList *p0 , *z ;

    for( p0 = top ; _____ ; p0 = p0->next ) {
        if ( _____ )
            break ;
    }
    z = _____
    if ( z != NULL ) {
        z->next = _____ ;
        z->prev = _____ ;
        _____
    } _____
}
```

4 説明問題

1. 設問2のようなデータ構造をデータの順序を意味する言葉を用いて一般的に何というか？
2. 設問2のようなデータ構造をリスト構造でなく、配列を使って実装する場合には、一般的にどのような方法が用いられるか、具体的に説明せよ。
3. 設問3の初期状態で入れられた-999,999のような目的のデータを一般的に何というか？
4. 設問3のようなデータ構造を使う場面とはどのような場合か？具体的な実装例を上げ、そのデータに向いている理由を説明せよ。

5 説明問題

リスト構造と配列を比較した場合の利点と欠点を、処理速度のオーダの式などを交えながら具体的に説明せよ。3つの点で利点 / 欠点を回答すること。

6 プログラム記述問題

くだもの買い物リストが、右図のように与えられる。リスト構造は、くだもの名前・値段・個数からできている。(例:apple,80円,4個...)

このデータ構造を扱うリスト構造 `struct FruitsList` を宣言し、そのリストの先頭を引数に与えたら、この買い物に必要なお金の合計を返り値とする関数

```
int fruitsCosts( struct FruitsList* p ) {...}
```

を作成せよ。

