

以下の6つ設問から5つを選んで回答せよ。

1 再帰動作トレース (20)

```
int find( struct List* p , int key ) {
    for( ; p != NULL ; p = p->next ) {
        printf( "%d-%d\n" , p->data , key ) ;
        ~~~~~(A)
        if ( p->data == key )
            return 1 ;
    }
    return 0 ;
}

int set_uniq( struct List* p ) {
    if ( p == NULL )
        return 1 ;
    else if ( find( p->next , p->data ) )
        return 0 ;
    else
        return set_uniq( p->next ) ;
}

void main() {
    struct List* top = cons( 1 , cons( 2 , cons( 3 , cons( 3 , NULL ) ) ) ) ;
    if ( !set_uniq( top ) )
        printf( "uniq\n" ) ;
}

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 ;
}

-[設問]-----
このリスト処理のプログラムで、
表示される内容を答えよ。(3x6+2)
-----
```

uniq
not-uniq

2 説明問題 (20)

- 2進数を用いた集合計算をするための方法を説明せよ。集合の要素は0~30までの整数とし、 $A=\{1,2,3\}, B=\{3,5,7,9\}$ で、 $A \cap B, A \cup B$ をどのように計算するか、具体的な値を交えて説明せよ。(12)
- 上記の計算方法で、差集合 $A - B = \{1,2\}$ の計算方法を説明せよ。(8)

3 データイメージ図(20)

以下のようなプログラムを作成した。設問に答えよ。

```
int stack[ 100 ] ;      | struct List* queue = NULL ;
int* sp = stack ;      | struct List** tail = &queue ;
                        | ~~~~~(A)
void push( int x ) {   | void put( int x ) {
    * sp = x ;         |     *tail = cons( x , NULL ) ;
    ~~~(B)             |     tail = &( (*tail)->next ) ;
    sp++ ;             | }
}                       | ~~~~~(C)
int pop() {           | int get() {
    sp-- ;             |     struct List* d = queue ;
    return *sp ;      |     int ans = d->data ;
}                       |     queue = queue->next ;
                        |     ~~~~~(E)
                        |     free( d ) ;
                        |     return ans ;
                        | }
void main() {         | } [2]
    push( 1 ) ;
    push( 2 ) ;
    printf( "%d\n" , pop() ) ; /* (F) */ (A)-----
    push( 3 ) ;
    put( 11 ) ;        (B)-----
    put( 22 ) ;
    printf( "%d\n" , get() ) ; /* (G) */ (C)-----
    put( 33 ) ;
}                       | ~~~~~(H) (D)-----
                        |
[1] (F)----- (G)----- (E)-----
```

1. 行(F),(G)の出力結果を答えよ。(2x2)
2. 下線部(A)~(E)の型を答えよ。(2x5)
3. 下線(H)の処理終了後のメモリの状態を図を用いて示せ。
図の解答は余白部に記入。(2x3)

4 説明問題(20)

1. 「配列は だがリストは である」といった対比した利点欠点を、2つ答えよ。(10)
2. 循環リストの構造と用途を、図や簡単な事例を交えて説明せよ。(10)

5 リスト処理の穴埋め

リスト構造を使った集合差を計算するプログラムを作りたい。以下の下線部を埋めよ。

```
~~~~~ set_diff( _____ a, _____ b ) {  
  (A) _____ (B) _____ (C)  
  struct List* ans = _____ (D)  
  for( ; _____ ; _____ ) {  
    (E) _____ (F)  
    struct List* p ;  
    for( p = _____ ; p != NULL ; p = p->next ) {  
      (G) _____  
      if ( _____ )  
        (H) _____  
      break ;  
    }  
    if ( p _____ NULL )  
      (I) _____  
    ans = _____ (J)  
  }  
  return ans ;  
}  
void main() {  
  struct List* a = cons( 1 , cons( 2 , cons( 3 , NULL ) ) ) ;  
  struct List* b = cons( 3 , cons( 5 , cons( 7 , NULL ) ) ) ;  
  print( set_diff( a , b ) ) ; // print() はリスト全要素表示関数  
}
```

6 課題理解確認 (20)

3 科目のテストの点数 (max100) と名前のデータをリスト構造で保存し、以下のような処理を行いたい。

各データ 1 人毎の最低点を求め、各人の最低点が全員の中で最低の人の名前を求めたい。ただし点数欄には、テストを受けなかった場合には-1 が保存されている。この例では、tsaitoh を返すこと。john-doe は全科目を受けていないので、該当しない。

次のような使い方をして、名前を表示できること。リストのデータを生成する処理は記述しなくてよい。

```
printf( "%s\n" , min_min_user( top ) ) ;
```

