問題と解答例（森下）

１　n 個の元から m 個を選択する場合の数 $_nC_m$ を計算するプログラムを記述せよ。

```
Public static int combin(int n, int m){
        if(n < m || n < 0 || m < 0){ return -1; }
        int x = 1;
        for(int i=0; i<m; i++){ x = x*(n-i); }
        for(int i=0; i<m; i++){ x = x/(m-i); }
        return x;
}
```

２　10 個の元からなる配列を Insertion Sort により整列化することを考える。比較の回数が最も多くなる配列の例と、最も小さくなる配列の例を示せ。

最も大きくなる例　JJHGFEDCBA　毎回、選択される元が先頭まで移動するから

最も小さくなる例　ABCDEFGHIJ　ソート済みなので、元の交換が発生しないから

３ リンクを使ってキューを実装するプログラムを記述せよ。

```
public class Queue {
        private Node head, tail;          // Node は講義ノートから借用
        public Queue(){ head = null; tail = null; }
        public void put(char c){
                Node x = new Node(c);
                if(queueEmpty()){ tail = x; head = x;}  // キューが空のとき head, tail を新ノードに
                else{ tail.next = x; tail = x; }        // 一番最後に x を付加
        }
        public char get(){
                char r = ‘ ‘;                  // キューが空の時には空白をかえすように
                if(!queueEmpty()){
                        r = (char) head.key;      // 空でないなら先頭の値を r へ
                        if(head == tail){ head = null; tail = null; } // １つしか元がない場合
                        else{ head = head.next; }  // 元が複数存在する場合
                }
                return r;
        }
        public boolean queueEmpty(){
                return (head == null);
        }
}
```

４　自然数（１以上の整数と定義）をいくつかの自然数の和として表現する仕方の個数を分割数と呼ぶ。たとえば、２は２と１＋１の２通りに表現できるので、分割数は２。３は３と２＋１と１＋１＋１の３通り（２＋１の和の順番を変更した１＋２は１通りに勘定する）に表せるので、分割数は３である。

(1)　7 と 8 の分割数を求めよ。

(2)　与えられた自然数の分割数を計算するプログラムを記述せよ。

(1)　7 の分割数は 15, 8 の分割数は 22

(2)　n を i 個の自然数の和で表現する仕方を pn(n, i) 通りとする。 pn(n,n) は pn(n,1) どちらも 1 通り。1 < i < n のとき、分割された自然数のうち少なくとも一つは１である場合と、すべてが２以上の場合に分ける。前者は 1 + (n-1 を i-1 個に分割した和) の形をしているので分割数は pn(n-1, i-1)。後者は、n = x1 + x2 + ... + xi のように分割されていれば、n-i = (x1 - 1) + (x2 - 1) + ... + (xi -1) のような分割へと変形できるので、分割数は pn(n-i, i) である。この結果 n-i < i となる事態が起こる可能性がある。そこで n < i ならば pn(n, i) = 0 と約束する。以上のようにして

pn(n, 1) = 1, pn(n, n) = 1

pn(n, i) = pn(n-1, i-1) + pn(n-i, i) ただし 1 < i < n のとき

pn(n, i) = 0　ただし n < i のとき

pn(n) = pn(n, 1) + pn(n, 2) + ... + pn(n, n)

プログラムにすると

```
public static int pn(int n){
	int sum = 0;
	if(n > 0){
		for(int i=1; i<=n; i++){ sum += pn(n,i); }
	}
	return sum;
}
public static int pn(int n, int i){
	if(n < i){ return 0; }
	if(n == i || i == 1){ return 1; }
	return pn(n-i,i)+pn(n-1,i-1);
}
```