

■ 다음 문제들을 푸시오. (반드시 과정을 보일 것)

[Recurrence] 다음 Java 함수 f() 를 보고 물음에 답하십시오.

```
static int f(int n) {
    if ( n == 1 ) {
        return 1;
    }
    else {
        return (f(n/2) + 1);
    }
}
```

1. f(101) 의 값은? [1]
2. 산술연산 (+, -, * /), 비교연산 (==), 함수의 call, return 은, 각각 1 의 시간이 필요하며, 그 밖의 연산에 걸리는 시간은 무시한다고 하자. 1 보다 크거나 같은 주어진 정수 n 에 대하여 함수 f(n) 을 처리하는데 걸리는 시간을 T(n) 이라 할 때, T(n) 을 n 의 점화식 (recurrence equation) 으로 나타내시오. (n 은 항상 1 보다 크거나 같다.) [3]
3. n 이 2 의 승수, 즉 $2^0, 2^1, 2^2, 2^3, 2^4, \dots$ 의 형태의 값을 가질 때, 2 번의 점화식을 풀어 T(n) 의 일반식을 구하십시오. [4]

[List] 다음은 배열로 구현된 class ArrayList 의 일부 모습이다. 리스트의 반복자 (Iterator)가 필요하여 내부 클래스 (Inner Class)로 구현하여 추가하려고 한다.

```
public class ArrayList<E> {
    private int _capacity; // 리스트의 최대 크기
    private int _size; // 리스트의 현재 크기
    private E[] _elements; // 원소를 저장하는 배열
    .....

    public int capacity() {...}
    private void setCapacity (int newCapacity) {...}
    public int size() {...}
    private void setSize (int new Size) {...}
    private E[] elements() {...}
    private void setElements (E[] newElements) {...}

    public Iterator<E> iterator () {
        return ( new ListIterator() );
    }
    private class ListIterator implements Iterator<E>
    {
        ..... // 여기를 채우시오 (문제 4)
    } End of Inner Class ListIterator
} // End of Class List
```

ListIterator 가 구현하는 interface 의 공개함수는 다음의 두 개를 사용한다.

```
public interface Iterator<E> {
    public boolean hasNext ();
    // 다음 원소가 존재하는지를 알려준다
    public E next ();
    // 다음 원소를 얻는다.
    // 원소를 얻고 나면, 그 다음 원소가 새로운 다음 원소가
    // 되어야 한다. 다음 원소가 없으면 null 을 얻는다.
} // End of interface Iterator
```

4. ArrayList 의 Inner Class 인 ListIterator 를 현재 주어져 있는 상태에서 완성하십시오. 인스턴스 변수, 생성자 등 필요한 모든 것을 채우시오. 인스턴스 변수에 대해 getter/setter 를 만들지 말고 구현하십시오. 단 this 는 사용할 수 있는 곳에서 생략하지 마시오. [6]
5. ListIterator 를 사용하여 리스트의 모든 원소의 합계를 내어, 정수형 변수 sum 에 저장하는 코드를 다음과 같이 시작하

여 작성하십시오. [3]

```
public class Something {
    .....
    private void run ()
    {
        ArrayList<Integer> myList = new ArrayList<Integer>();
        ..... // 리스트에 원소가 적절히 삽입되는 코드
        // 이 시점에 리스트는 완성되어 있다고 가정한다.
        // 이어서, 문제에서 요구하는 코드를 작성하십시오.

        int sum = 0;
        ..... // 여기를 채우시오 (문제 5)
    }
}
```

[시험문제 끝]