

1. Given the following statements for the students' scores:

```
final int MAX = 21;    // # of students
int[] scores = new int[MAX];
...
```

Write a segment of code to read data from a file as follows, and store them to the above array. The number of data in the file is unknown. Declare the variable needed. Assume the *Scanner* object is *input*.

```
12
7
78
...
93
```

```
int count = 0;
while(input.hasNext())
{
    scores[count] = input.nextInt();
    count ++;
}
```

2. Based on array in the last question, write the segment of code to sum up all the data. Also find the largest and smallest.

```
// have the array from the above code
int sum = scores[0];
int max = scores[0];
int min = scores[100];
for(int i=1; i<count; i++)
{
    sum += scores[i];
    if(max < scores[i])
        max = scores[i];
    if(min > scores[i])
        min = scores[i];
}
```

OR

```
// not have the array
```

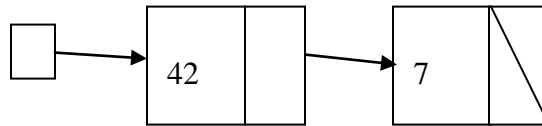
```
int count = 0;
int sum = 0;
int max = 0;    // assume data range: 0 - 100
int min = 100;
while(input.hasNext())
{
    scores[count] = input.nextInt();
    sum += scores[count];
    if(max < scores[count])
        max = scores[count];
    if(min > scores[count])
        min = scores[count];
    count ++;
}
```

3. Given the following definition of the integer list node.

```
public class ListNode
{
    private int data;
    private ListNode link;
    public ListNode();
    public ListNode(int d, ListNode l);
    public void setData(int d);
    public void setLink(ListNode l);
    public int getData();
    public ListNode getLink();
}
```

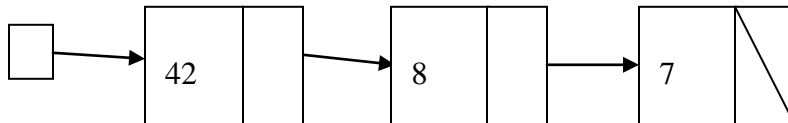
Based on the above integer list node class, write the statement(s) to perform the following operations. Declare any variables and objects needed.

a) Create the following node.



```
ListNode rear = new ListNode(7, null);
ListNode temp = new ListNode(42, rear);
```

b) Insert a node to the middle of the above nodes.



```
ListNode mid = new ListNode(8, rear);
temp.setLink(mid);
```

OR:

```
ListNode mid = new ListNode(8, null);
mid.setLink(rear);
temp.setLink(mid);
```

c) Make the above list into a circular list.

```
rear.setLink(temp );
```