C-3.17: ANSWER

The solution:

Use two loops. In the outer loop, pick elements one by one and count the number of occurrences of the picked element in the inner loop.   
This method doesn’t use the other useful data provided in questions like range of numbers is between 1 to n and there are only two repeating elements.

C-3.18: ANSWER

The solution:

The idea is to use nested loop and for each element check if the element is present in the array more than once or not.

C-3.19: ANSWER

The solution:

Public GameEntry {

private String name ;

private int score;

public add(String n, int s)

{

Name=n;

Score=s;

}

public String getName() { return name; }

public String getScore() { returnScore; }

public String toString() { name+" "+Score}

}

public void add(GameEntry e)

{

int newScore=e.getScore;

if (numEntries<board.length || newScore> board [numEntries-1].getScore())

{

If(numEntries<board.length)

numEntries++;

j--;

}

int j= numEntries-1;

While (j>0 && board [j-1].getScore()< newScore)

{

board [j]= board[j-1];

}

board [j]=e;

}

public GameEntry remove(int i) throws IndexOutOfBoundsExecption {

if(i<0 || i>=numEntries)

throw new IndexOutOfBoundsExecption("invalid index "+i);

GameEntry temp=board[i];

For (j=i ; j<numEntry-1; j++)

board [j]= board[j+1];

board [numEntry-1]=null;

numEntry--;

return temp;

}

public class ScoreBoard {

private int numEntries=0;

private GameEntry[] Board;

public ScoreBoard (int capacity)

{

Board= new GameEntry[capacity];

}

}