

SKILL- 5

ID.NO: 2100033187

Sec: 13

Subject: AOO

1.

You are given a set S of N elements. You are supposed to form an array A such as: S

= {1,2,3} then we have its subsets as {},{1},{2},{3},{1,2},{2,3},{1,3},{1,2,3}.

Then A consists of the largest number(0 for empty set) from each subset.

A = {0,1,2,3,2,3,3,3}

Now the actual task comes here.

there are three types of queries such as :

m is an integer.

< m , no. of elements in A that are < m.

> m , no. of elements in A that are > m.

= m , no. of elements in A that are = m.

Code:

```
package skill5;
public class SetsandMaps
{ public static void main(String[] args)
    { int set[]={1,2,3}; int
      count=0,c=0,t=0,p=0,n=0; int
      arr[]={0,1,2,3,2,3,3,3};
      for(int i=0;i<8;i++)
      { if(arr[i]==3)
        { count++;
        }
        if(arr[i]>3)
        { c++;
        }
        if(arr[i]==2)
        { t++;
        }
        if(arr[i]>1)
        { p++;
        }
        if(arr[i]<1)
        { n++;
        }
      }
      System.out.println(c);
      System.out.println(t);
      System.out.println(count);
      System.out.println(p);
      System.out.println(n);
    }
}
```

2. EXAM SETS

Given N question papers with their difficulty levels as A0,A1,A2....AN-1. A subset is something that includes two question papers with different difficulty levels. Each question paper will appear at most 1 time among all the subsets.Find number of subsets that can be made.

CODE:

```
import java.util.*

class TestClass { public static void main(String args[] )
    throws Exception {
    Scanner s = new Scanner(System.in);
    String name = s.nextLine();
    System.out.println("Hi, " + name + "."); try
    {
        Scanner s = new Scanner(System.in);
        int count = s.nextInt(); for (int i = 0; i <
        count; i++) {
            long magicItems = s.nextLong();
            long itemLimit = s.nextLong();
            long visits = s.nextLong(); if
            (visits == 0)
            {
                System.out.println(0);
            }
            Else
            {
                long mana = itemLimit >> (visits-1); if
                (visits > 58) {
                    System.out.println(magicItems);
```

```

    }
    else if (mana >= magicItems) {
        System.out.println(0);
    }
    else {
        System.out.println(magicItems - mana);
    }
}
catch (Exception e) {
    System.out.println(e);
}
}
}
}

```

Output:

The screenshot shows the HackerEarth interface for a problem titled "The exam". The problem description states: "Kate is wizard. Now she should pass the exam. During the exam she have to go into the magic room N times. Initially there are X magic items in the room. Before each visit she can use magic to decrease the number of magic items in the room. For each item she spends 1 mana point to destroy it. After each visit the number of magic items increases by 2 times. She can't go into room if there are more than L magic items in it. What minimum mana points does Kate need to pass the exam? You need to solve this problem for several test cases."

The input format is: "The first line of input contains single integer T ($1 \leq T \leq 10^5$) - a number of test cases."

The submission details show a score of 0, time of 1.33003 seconds, memory of 95296 KiB, and language of Java 14. The result is "Accepted".

| Input | Result | Time (sec) | Memory (KiB) | Score | Your Output | Correct Output |
|-------|----------|------------|--------------|-------|-------------|----------------|
| 65 | Accepted | 1.33003 | 95296 | 0 | | |

3. Attendance

Chef is teaching a cooking course. There are NN students attending the course, numbered 11 through NN .

Before each lesson, Chef has to take attendance, i.e. call out the names of students one by one and mark which students are present. Each student has a first name and a last name. In order to save time, Chef

wants to call out only the first names of students. However, whenever there are multiple students with the same first name, Chef has to call out the full names (both first and last names) of

all these students. For each student that does not share the first name with any other student, Chef may still call out only this student's first name.

Help Chef decide, for each student, whether he will call out this student's full name or only the first name.

CODE:

```
import java.util.*;

import java.lang.*;

import java.io.*;

class Codechef
{
public static void main (String[] args) throws java.lang.Exception {

    Scanner sc = new Scanner(System.in);

    int T = sc.nextInt();

    for(int z=0;z<T;z++){

        int N = sc.nextInt();

        String A[] = new String[N];

        String B[] = new String[N];

        for(int j=0;j<N;j++){

            A[j] = sc.next();

            B[j] = sc.next();

        }

        for(int k=0;k<N;k++){ int count

            =0; for(int l=0;l<N;l++){
```

```

        if(A[k].equals(A[l])){

            count++;

        }
        if(count>1){

            System.out.println(A[k]+" "+B[k]);

        }
    else{

        System.out.println(A[k]);

    }

    }

}

}

}

}

}

```

OUTPUT:

The screenshot shows a web browser window with the CodeChef website. The page is for a problem titled "Attendance - Submit". The problem statement describes a cooking course with N students. Chef wants to call out the names of students present, but only the first name if it is unique, or the full name if it is not. The input consists of T test cases. The first line of each test case is N , followed by N lines of student names. The output should be the names to be called out for each test case.

The submission section shows a successful submission with the status "Correct Answer", submission ID "71508920", and a time of "0.39s". The subtask score is "100.00%" and the result is "AC (0.392774)".

| Sub-Task | Task # | Result (time) |
|----------|--------|---------------|
| 1 | 0 | AC (0.392774) |

Subtask Score: 100.00% Result - AC