**Task 1:-**

**Task: Take input as integer array , and find out the nth biggest and mth lowest number from that. Eg: 1, 7, 2, 9, 12, 17, 20, 34, 21 and n= 3, m=4 then biggest is: 20 and lowest is: 9**

**Program:-**

**package** practice.tasks;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** Task1 {

**public** **static** StringBuffer findMaxMin(**int** n,**int** a[],**int** size,**int** m)

{

StringBuffer sb=**new** StringBuffer();

Arrays.*sort*(a);

**int** maxLoc=a.length-n;

**int** minLoc=m-1;

**for**(**int** i=0;i<a.length;i++)

{

**if**(i==maxLoc)

sb.append("Biggest: "+a[i]);

**if**(i==minLoc)

sb.append("Lowest: "+a[i]+" ");

}

**return** sb;

}

**public** **static** **void** main(String[] args) {

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("enter the size:");

**int** size=in.nextInt();

System.***out***.println("Enter the elements of array");

**int** a[] = **new** **int**[size];

**for**(**int** i=0;i<a.length;i++)

a[i]=in.nextInt();

System.***out***.println("Enter nth biggest number");

**int** n=in.nextInt();

System.***out***.println("Enter mth lowest number");

**int** m=in.nextInt();

System.***out***.println(Task1.*findMaxMin*(n, a,size,m));

}

}

**Task-2:**

**Task: Take input as multiple strings, and arrange the strings in the order by the following constraints: 1) sum with highest ascii count 2) highest character existence**

**Program:-**

**(Task2.java)**

**package** practice.tasks;

**import** java.util.HashSet;

**import** java.util.Scanner;

**import** java.util.TreeMap;

**public** **class** Task2 {

**public** **static** **void** main(String[] args) {

Scanner in = **new** Scanner(System.***in***);

TreeMap<StringCk, String> tm = **new** TreeMap<StringCk, String>(**new** ArrangeComparator());

System.***out***.println("Enter the no. of strings:");

**int** size=in.nextInt();

String str[]= **new** String[size];

System.***out***.println("Enter the strings:");

**for**(**int** i=0;i<size;i++)

str[i]=in.next();

**for**(**int** i=0;i<str.length;i++)

{

**int** asciiCount=0;**int** charCount=0;

StringCk sck = **new** StringCk();

HashSet<Character> hs = **new** HashSet<Character>();

**for**(**int** j=0;j<str[i].length();j++)

{

asciiCount+=str[i].charAt(j);

**if**(hs.contains(str[i].charAt(j)))

charCount++;

**else**

hs.add(str[i].charAt(j));

}

System.***out***.println(str[i]+" "+asciiCount+" "+charCount);

sck.setAsciiCount(asciiCount);

sck.setCharCount(charCount);

tm.put(sck, str[i]);

}

System.***out***.println(tm);

}

}

**(StringCk.java)**

**package** practice.tasks;

**public** **class** StringCk {

**int** asciiCount,charCount;

**public** **int** getAsciiCount() {

**return** asciiCount;

}

**public** **void** setAsciiCount(**int** asciiCount) {

**this**.asciiCount = asciiCount;

}

**public** **int** getCharCount() {

**return** charCount;

}

**public** **void** setCharCount(**int** charCount) {

**this**.charCount = charCount;

}

}

**(ArrangeComparator.java)**

**package** practice.tasks;

**import** java.util.Comparator;

**public** **class** ArrangeComparator **implements** Comparator<StringCk>{

@Override

**public** **int** compare(StringCk o1, StringCk o2) {

**if**(o1.getAsciiCount()<o2.getAsciiCount())

{

**if**(o1.getCharCount()<o2.getCharCount())

**return** 1;

**else** **if**(o1.getCharCount()>o2.getCharCount())

**return** -1;

**else**

**return** 0;

}

**else** **if**(o1.getAsciiCount()>o2.getAsciiCount())

{

**if**(o1.getCharCount()<o2.getCharCount())

**return** 1;

**else** **if**(o1.getCharCount()>o2.getCharCount())

**return** -1;

**else**

**return** 0;

}

**else**

{

**if**(o1.getCharCount()<o2.getCharCount())

**return** 1;

**else** **if**(o1.getCharCount()>o2.getCharCount())

**return** -1;

**else**

**return** 0;

}

}

}

**Sir Please Check the 2nd Task once I feel it’s correct…**