

TASK3.1:

```
import java.util.*;
import java.util.stream.Collectors;
class Student {
    String name;
    int marks;
    Student(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        List<Student> students = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            String name = sc.next();
            int marks = sc.nextInt();
            students.add(new Student(name, marks));
        }
        int k = sc.nextInt();
        List<String> topStudents = students.stream()
            .sorted(
                Comparator.comparingInt((Student s) -> s.marks)
                    .reversed()
                    .thenComparing(s -> s.name)
            )
            .limit(k)
            .map(s -> s.name)
            .collect(Collectors.toList());
    }
}
```

```
        System.out.println(String.join(" ", topStudents));  
    }  
}
```

Output

```
6
```

```
Arun 85
```

```
Bala 92
```

```
Charan 85
```

```
Divya 95
```

```
Esha 92
```

```
Farhan 88
```

```
3
```

```
Divya Bala Esha
```

```
} === Code Execution Successful ===
```

TASK3.2":

```
import java.util.Scanner;  
public class Solution {
```

```
public static String getSmallestAndLargest(String s, int k) {  
    String smallest=s.substring(0,k);  
    String largest=s.substring(0,k);  
    for(int i=0;i<=s.length()-k;i++){  
        String current=s.substring(i,i+k);  
        if(current.compareTo(smallest)<0){  
            smallest=current;  
        }  
        if(current.compareTo(largest)>0){  
            largest=current;  
        }  
    }  
    return smallest+"\n"+largest;  
}  
  
public static void main(String[] args) {  
    Scanner scan = new Scanner(System.in);  
    String s = scan.next();  
    int k = scan.nextInt();  
    scan.close();  
  
    System.out.println(getSmallestAndLargest(s, k));  
}  
}
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin)

[Download](#)

```
1 welcometojava  
2 3
```

Your Output (stdout)

```
1 ava  
2 we1
```

Expected Output

[Download](#)

```
1 ava  
2 we1
```

TASK3.3:

```
import java.util.*;  
  
class Checker implements Comparator<Player>{  
  
    public int compare(Player a,Player b){  
  
        if(a.score != b.score){  
  
            return b.score-a.score;  
        }  
  
        return a.name.compareTo(b.name);  
    }  
}  
  
class Player{  
  
    String name;  
  
    int score;  
  
    Player(String name, int score){  
  
        this.name = name;  
  
        this.score = score;  
    }  
}
```

```

class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        Player[] player = new Player[n];
        Checker checker = new Checker();
        for(int i = 0; i < n; i++){
            player[i] = new Player(scan.next(), scan.nextInt());
        }
        scan.close();
        Arrays.sort(player, checker);
        for(int i = 0; i < player.length; i++){
            System.out.printf("%s %s\n", player[i].name, player[i].score);
        }
    }
}

```

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Your Output (stdout)

```

1 aleksa 150
2 amy 100
3 david 100
4 aakansha 75
5 heraldo 50

```

Expected Output

```

1 aleksa 150
2 amy 100
3 david 100
4 aakansha 75
5 heraldo 50

```

[Download](#)

TASK3.4:

```
import java.util.*;

class Student{
    private int id;
    private String fname;
    private double cgpa;
    public Student(int id, String fname, double cgpa) {
        super();
        this.id = id;
        this.fname = fname;
        this.cgpa = cgpa;
    }
    public int getId() {
        return id;
    }
    public String getFname() {
        return fname;
    }
    public double getCgpa() {
        return cgpa;
    }
}

class StudentComparator implements Comparator<Student>{
    public int compare(Student s1, Student s2){
        if(s1.getCgpa()!= s2.getCgpa()){
            return Double.compare(s2.getCgpa(),s1.getCgpa());
        }
        int nameCompare=s1.getFname().compareTo(s2.getFname());
        if(nameCompare!=0){
            return nameCompare;
        }
    }
}
```

```
        return Integer.compare(s1.getId(),s2.getId());  
    }  
}  
  
public class Solution  
{  
  
    public static void main(String[] args){  
  
        Scanner in = new Scanner(System.in);  
  
        int testCases = Integer.parseInt(in.nextLine());  
  
  
        List<Student> studentList = new ArrayList<Student>();  
  
        while(testCases>0){  
  
            int id = in.nextInt();  
  
            String fname = in.next();  
  
            double cgpa = in.nextDouble();  
  
  
            Student st = new Student(id, fname, cgpa);  
            studentList.add(st);  
  
  
            testCases--;  
        }  
  
        Collections.sort(studentList, new StudentComparator());  
  
  
        for(Student st: studentList){  
            System.out.println(st.getFname());  
        }  
    }  
}
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test

Sample Test case 0

1	5
2	33 Rumpa 3.68
3	85 Ashis 3.85
4	56 Samiha 3.75
5	19 Samara 3.75
6	22 Fahim 3.76

Your Output (stdout)

1	Ashis
2	Fahim
3	Samara
4	Samiha
5	Rumpa

TASK3.5

```
import java.io.*;
import java.util.*;

class UserMainCode
{
    public String addNumberStrings(String input1, String input2)
    {
        int i = input1.length() - 1;
        int j = input2.length() - 1;
        int carry = 0;
        StringBuilder result = new StringBuilder();
        while (i >= 0 || j >= 0 || carry > 0)
```

```

    {
        int sum = carry;
        if (i >= 0)
            sum += input1.charAt(i--) - '0';
        if (j >= 0)
            sum += input2.charAt(j--) - '0';
        result.append(sum % 10);
        carry = sum / 10;
    }
    return result.reverse().toString();
}

[Running] cd "c:\Users\bhask\OneDrive\Desktop\week3\" && javac UserMainCode.java && java UserMainCode
1898
[Done] exited with code=0 in 1.639 seconds

```

```

class Solution {

    public String[] sortPeople(String[] names, int[] heights) {
        int n = names.length;
        // Create an array of indices
        Integer[] indices = new Integer[n];
        for (int i = 0; i < n; i++) {
            indices[i] = i;
        }

        // Sort indices by corresponding heights in descending order
        Arrays.sort(indices, (a, b) -> heights[b] - heights[a]);

        // Build the result array
        String[] result = new String[n];
        for (int i = 0; i < n; i++) {

```

```
    result[i] = names[indices[i]];  
}  
  
return result;  
}  
}
```

The screenshot shows a dark-themed IDE interface with two tabs at the top: "Case 1" and "Case 2". The "Case 1" tab is active, indicated by a blue border around its button. Below the tabs, there are two sections: "Input" and "Output".

Input:

```
names =  
["Mary", "John", "Emma"]
```

Output:

```
["Mary", "Emma", "John"]
```

Expected:

```
["Mary", "Emma", "John"]
```

TASK3.6:

```
class UserMainCode
{
    public int NthPrime(int input1)
    {
        if (input1 <= 0)
            return 0;

        int count = 0;
        int num = 1;

        while (count < input1)
        {
            num++;
            boolean isPrime = true;

            for (int i = 2; i * i <= num; i++)
            {
                if (num % i == 0)
                {
                    isPrime = false;
                    break;
                }
            }

            if (num >= 2 && isPrime)
                count++;

        }

        return num;
    }
}
```

Time: 125 ms

Memory: 57820 kb

</> TEST CASE INFORMATION

Input

15

Expected Output

47

Actual Output

47

TASK3.7:

```
import java.io.*;  
import java.util.*;  
  
class UserMainCode {  
    public int seriesN(int a, int b, int c, int N) {  
        if (N == 1) return a;  
        if (N == 2) return b;
```

```
if (N == 3) return c;

int gap1 = b - a;
int gap2 = c - b;
int current = c;

for (int i = 4; i <= N; i++) {
    if (i % 2 == 0) {
        current += gap1;
    } else {
        current += gap2;
    }
}

return current;
}
```

⌚ CODE EXECUTION DETAILS

Time: 132 ms

Memory: 57820 kb

</> TEST CASE INFORMATION

Input

1,3,6,100

Expected Output

248

Actual Output

248

}

TASK3.8:

```
public int AddSub(int input1, int input2) {  
    int N = input1;  
    int opt = input2;
```

```
int result = N;
boolean add;
if (opt == 1) {
    add = false;
} else {
    add = true;
}
for (int i = N - 1; i >= 1; i--) {
    if (add) {
        result += i;
    } else {
        result -= i;
    }
    add = !add;
}
return result;
```

 default

 CODE EXECUTION DETAILS

Time: 128 ms

Memory: 57820 kb

 TEST CASE INFORMATION

Input

10000,2

Expected Output

15000

Actual Output

15000

 CONSOLE OUTPUT

}