1. Write a Java program which performs the following operations.

1. Bitwise AND of two numbers
2. Bitwise AND of three numbers

Implement **compile time polymorphism** by assigning the same function name with a different number of arguments.

Input Format:
Input is given as space separated list of numbers (maximum 3 numbers)

Output Format:
Single Integer denoting the output of the operation.

Sample Input 1:
5 12

Sample Output 1:
4

Sample Input 2:
9 7 17

Sample Output 2:
1

2. Write a Java program to calculate the area of different shapes: circle, rectangle and triangle. You should use the function overloading concept in your program to automatically identify the shape of the object and to calculate the area corresponding to that object. The circle needs a single integer parameter (radius) , the rectangle needs two integer parameters (length, breadth) and the triangle needs three integer parameters(side a, side b, side c).

Input Format:
Input is given as space separated list of numbers(maximum 3 numbers)

Output Format:
<Name of the shape>: <Area rounded to two decimal places>

Sample Input 1:
4 2

Sample Output 1:
Rectangle:8.00

Sample Input 1:
3

Sample Output 1:
Circle:28.27

3. In a school there are “n” students standing in a line. You have to create a Student class which has 3 attributes : name, rollNo, height. The students are standing according to their rollNo.(increasing order). PT teacher wants to select “k” students among them so that the selected students are in **strictly increasing height**. Now you have to help the PT teacher so that he can get the list of students. If there is more than one student having the same height, choose the student who comes first in the Alphabetical order of their names.

Input Format:
First line specifies the number of students. Then from next line onwards, student details in format of (Name,Roll No.,Height) line by line.

Output Format:
First line shows the integer representing the number of students selected
Student names separated by comma.

Sample Input1:
```
4
(Abhay,1,10)
(Soham,2,20)
(Jeet,3,10)
(Akshay,4,40)
```

Sample Output1:
```
3
Abhay,Soham,Akshay
```

Sample Input2:
```
9
(Abhay,1,10)
(Soham,2,22)
(Jeet,3,10)
(Akshay,4,33)
(Riya,5,33)
(Abhishek,6,50)
(Vidit,7,50)
(Disha,8,60)
(Lakshya,9,80)
```

Sample Output2:
```
6
```

Abhay,Soham,Akshay,Abhishek,Disha,Lakshya

4. Write a java program with a class Customer with properties: customerName, accountNo, currentBalance, and member function: availableBalance (displays <customerName>:<currentbalance >). Create subclasses for Customer, based on memberships type- Silver, Gold, Platinum. In each subclass, override the function availableBalance(), which displays current balance and the predicted amount of balance after 10years (displays <customerName>:<current balance >:<Membership type>:<predicted balance>).

SilverMembership: Customers have balances less than one lakh, and the bank provides 5% of rate of interest to their silver customers.
GoldMembership: Customers have a balance of more than one lakh and less than ten lakhs and the bank provides 10% of rate of interest to their gold customers.
PlatinumMembership: If customer has balance more than ten lakhs and bank provides 15% of rate of interest to their platinum customers.

Note: The type (datatype of reference variable) of the array should be Customer (Eg: Customer listOfCustomers [n]).

Input Format:

The first line contains a single integer n, the number of customer's details stored.
The next n line contains following colon-separated values:
- Customer name
- Customer account number
- Current Balance

Output Format:

Output contains n line with following colon-separated values
- Customer name
- Current Balance
- Membership type
- Predicted balance

Sample Input:
```
2
Suresh:LD703:500000
Dinesh:LD905:1500000
```

Sample Output:

```
Suresh:500000:Gold:550000
Dinesh:1500000:Platinum:1725000
```

5. Write a java program with a class **Car** with properties ***modelName***, ***engineNo***, ***basePrice***, ***additionalCharge*** and a function ***totalPrice*** which returns the sum of basePrice and additionalCharge. Create a **Dealer** class which contains ***dealerName*** (String), ***discountPercentage***(float) and a Car object as attributes. Member function ***onRoadPrice*** in Dealer class returns the dealer price of the car after applying a discount percentage on the total price of the Car.

Input Format:

The first line contains a single integer n, the number of cars whose details are stored.
The next n line contains following space-separated values:

- Model name
- Engine number
- Base price
- Additional charges
- Dealer name
- Discount Percentage

Output Format:

Output contains the following space-separated car details:

- Model name
- Total price
- On Road price

Sample Input:

```
1
Mahendra  cc2300  1000000  50000 KVR 10
```

Sample Output:

```
Mahendra 1050000  945000
```

Sample Input:

```
1
Hyundai  cc3400  500000 20000 HondaDrive 8
```

Sample Output:

Hyundai 520000 478400

6. Create a class **Employee** with attributes *employeeId*, *employeeName* and *salary*. Create a method *CalculateSalary()* in Employee Class. Create a class **PermanentEmployee** which extends the **Employee** class and has the attributes *basicPay* and *PFAmount*. Override the *CalculateSalary()* method in PermanentEmployee class as given below

salary = basic_pay - PF_amount ; where PF_amount is 12% of basic_pay

Create another class **Temporary Employee** which extends from **Employee** with attributes *hoursWorked* and *hourlyWages***.** Implement the CalculateSalary() method in TemporaryEmployee as:

salary = hourlyWages * hoursWorked

Now display the salary of the Employee given his employeeId depending on the type of Employee. If there is no employee with a given employeeId return '-1'.

Input Format:

First line contains the number of employees 'N'.

Followed by N lines containing the details of each employee. Details of each employee is given as space separated list of (employee_type, employeeId, employeeName, [parameter list depending employee_type])
Integer '1' denotes Permanent Employee, and '2' denotes Temporary employee.
If the employee_type is '1' then his basic_pay is given as input.
If the employee_type is '2' then hoursWorked and hourlyWages are given as input.

Last line contains an employeeId for which you have to display his salary.

OutPut Format:

Output contains details of the employee (employeeId, employeeName, salary) with the given employeeId separated by space

Sample Input 1:
```
3
1 203 shiva 15000
2 312 krishna 43 250
1 415 ankit 25000
312
```

Sample Output 1:
```
312 krishna 10750
```

Sample Input 2:
```
4
2 672 messi 55 25000
```

Sample Output 2:
```
326 neymar 29040
```

```
1 326 neymar 33000
2 112 mbappe 38 23000
1 724 ramos 35000
326
```

7. Let PGStudent be a subclass of Student, where Student is a subclass of Person. A person has *personName* and *age*; a student has *rollNumber* and *percentageOfMarks*. The PGStudent has *elective* and *numberOfSubjects*. Write a java program to read details of N number of PG students and show their names in the order of *percentageOfMarks*. That is, display the Name of top scorer first and then second topper and so on.

Input Format:
First line contains the number of students, N.
The next N lines are the student details separated by colon(Name: Age: Elective: Percentage) in each line

Output Format:
Names of students (line by line) in the descending order of their Percentage

Sample input:
```
3
Amal:22:Soft Computing:77
Neha:21: IOT:75
Kiran:21:Image Processing: 80
```

Sample output:
```
Kiran
Amal
Neha
```