Q1. Write a java program that reads N integers into an array and copies the integers from a given starting index ‘*i*’ to an ending index ‘e’ into a new array. On the newly copied array, perform the following operations:

- Find two elements in the copied array such that the difference between them is the largest. *Note: the order of these two elements should be the same as they appear in the array.*
- Find the sum of the elements in the new array.
- Check if the sum of elements in the new array is divisible by a given number ‘*d*’.

The program should handle following exceptions

- Print “***FormatError***” if *“InputMisMatchException”* occurs.
- Print “***ArrayIndexError***” if **“***ArrayIndexOutOfBoundsException”* or *“NoSuchElementException*” occurs.
- Print “***ArithmeticError***” if *“DividebyZero”* occurs.

Input format

First line has the size of the array N.
The next N lines have the N integers. The next two lines will be in the following format:
i <value> e <value> (Here, ***i*** *is the starting index and* ***e*** *is the ending index of the array to be copied)*
d<value> (To check whether the sum of the new array is divisible by <value>)

Output Format

First line prints the input array
Second line prints the copied array
Third line prints the two elements such that the difference between them is largest.
Fourth line prints sum of the elements in the copied array
Fifth line prints YES or NO indicating whether the sum is divisible by the given number.

6 3 5 1 4 2 6 i 2 e 4 d 3	3 5 1 4 2 6 5 1 4 5 1 10 NO
6 22 45 3 34 11 21 i 2 e 7	22 45 3 34 11 21 ArrayIndexError

4 55 70 12 5.5	FormatError

Q2. Write a java program that will read a student academic credit data and create a list of students on academic warning. Each line of the input will contain the *student name* (a single String without spaces), the *number of semester hours* (an integer), the *total quality points* (a double). The program should compute the grade point for each student and determine if the student is on academic warning

Grade Point (GP) = total quality points divided by the number of semester hours (rounded to two decimal places)

A student will be on academic warning if

- **GP** is less than 1.5 for students with fewer than 30 semester hours
- **GP** is less than 1.75 for students with semester hours above or equal to 30, but fewer than 60
- **GP** is less than 2.0 for all other students.

Implement exception handling mechanism to handle the following cases:

- “***FormatError***” if *“NumberFormatException”* occurs.
- “***NegativeInput***” if *input contains negative values*.
- “***Overlimit***” if *no of semester hours is larger than* 100.
- “***ArithmeticError***” if *“DividebyZero”* occurs.

In all the above cases, print the name of the student whose data entry encountered an exception as specified in the output format.

Input Format

First line contains the number of student data, **N**.
Each line of the input is three space separated values
<String> <Integer> <Double>

Output Format

Output line contains the name of the students on academic warning, separated by a newline
In case of exception

- “***FormatError:<studentname>***” if *“NumberFormatException”* occurs.
- “***NegativeInput:<studentname>***” if *input contains negative values*.
- “***Overlimit:<studentname>***” if *no of semester hours is larger than* 100.
- “***ArithmeticError:<studentname>***” if *“DividebyZero”* occurs.

TEST CASES

7 Smitha 27 83.7 James 21 28.35 Deepak 60 150 Renju 96 182.4 Jeeva 100 400 Arun 33 57.4 Melvin 83 190	James Renju Arun
3 Abin 70.56 198 Arun 33 57.4 Melvin 83 190	FormatError:Abin
2 Vishal 84 168 Akhil 0 83.2	ArithmeticError:Akhil

Q3. Create an **Inventory** class with the data members **stockNumber** ***(Integer),*** **quantity** ***(Integer)***, and **price** ***(Double)***, and the methods **dataEntry** and **display.** The **dataEntry** method will read all the inventory details and finally the display method displays the inventory details as shown in output format. The **dataEntry** method throws exception in the following cases:

- If the **stockNumber** is negative or higher than 999 print ***InvalidStockNumber***
- If the **quantity** is less than or equal to 0 print ***InvalidQuantity***
- If the **price** is greater than 100.0 print ***InvalidPrice***

When an exception is detected, dataEntry method will only print the exception and terminate the program.

Input Format

First line contains the number of inventory objects, **N**.
Each of the **N** lines of input is three space separated values <**stockNumber> <quantity> <price>**

Output Format

Print the data fields of all the objects in the following format

*Stocknumber:<**stockNumber**>*

*Quantity:<**quantity**>*

*Price:<**price**> (rounded to 1 decimal places)*

TEST CASES

INPUT	**OUTPUT**

2
1 20 80.00
2 30 92.00
	Stocknumber:1
	Quantity:20
	Price:80.0
	Stocknumber:2
	Quantity:30
	Price:92.0

2
4 50 63.00
-19 10 25.00
	InvalidStockNumber