Crie um programa onde você receberá valores inteiros não negativos como entrada.

Ordene estes valores de acordo com o seguinte critério:

* Primeiro os Pares
* Depois os Ímpares

Você deve exibir os pares em ordem crescente e na sequência os ímpares em ordem decrescente.

**Entrada**

A primeira linha de entrada contém um único inteiro positivo **N**(1 < **N** < 10000) Este é o número de linhas de entrada que vem logo a seguir. As próximas **N** linhas terão, cada uma delas, um valor inteiro não negativo.

**Saída**

Exiba todos os valores lidos na entrada segundo a ordem apresentada acima. Cada número deve ser impresso em uma linha, conforme exemplo de saída abaixo.

| Exemplo de Entrada | Exemplo de Saída |
| --- | --- |
| 10 4 32 34 543 3456 654 567 87 6789 98 | 4 32 34 98 654 3456 6789 567 543 87 |

Código

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import java.util.ArrayList;

import java.util.Scanner;

import java.util.Collections;

/\*\*

\*

\* @author Dougl

\*/

public class Divisao {

/\*\*

\* @param args the command line arguments

\*/

static ArrayList<Integer> vetorPar = new ArrayList();

static ArrayList<Integer> vetorImpar= new ArrayList();

static ArrayList<Integer> parImpar= new ArrayList();;

public static void main(String[] args) {

// TODO code application logic here

Scanner ler = new Scanner(System.in);

int num = ler.nextInt();

while(ler.hasNext()) {

num = ler.nextInt();

if(num % 2 == 0)

vetorPar.add(num);

else

vetorImpar.add(num);

}

Collections.sort(vetorPar);

Collections.sort(vetorImpar);

Collections.reverse(vetorImpar);

parImpar.addAll(vetorPar);

parImpar.addAll(vetorImpar);

for(int i = 0; i < parImpar.size(); i++)

System.out.println(parImpar.get(i));

}

}