

Во рамки на една еднострано поврзана листа се чуваат податоци за студенти.

Во секој јазол од чистата се чува индекс, име и поени за студентот.

Потребно е од листата да се отстранат (избришат) сите студенти за коишто е точно дека имаат најмалку поени.

Постои само еден таков студент во листата.

ВЛЕЗ:

5

151615 Jessica 55

161718 Joe 89

161719 Mike 79

121315 Luke 80

131467 Peter 90

ИЗЛЕЗ:

Joe

Mike

Luke

Peter

Vlez:

3

161501 Peter 100

171010 Sarah 50

181018 Joe 70

Izlez:

Peter

Joe

Vlez:

3

161515 Joe 56

131010 Mark 78

141517 Emily 99

Izlez:

Mark

Emily

4

161414 Luke 100

151515 Sarah 89

131414 Emily 99

171717 Tom 67

Izlez:

Luke

Sarah

Emily

Vlez:

3

161515 Joe 56

131010 Mark 78

141517 EMily 99

Izlez:

Mark

Emily

Inicijalen kod:

import java.util.Iterator;

import java.util.NoSuchElementException;

import java.util.Scanner;

class SLLNode {

String index;

String name;

int points;

SLLNode succ;

public SLLNode(String index, String name, int points, SLLNode succ) {

this.index = index;

this.name = name;

this.points = points;

this.succ = succ;

}

@Override

public String toString() {

return name;

}

}

class SLL {

SLLNode first;

public SLL() {

this.first = null;

}

public void insertFirst(String index, String name, int points) {

SLLNode ins = new SLLNode(index, name, points, first);

first = ins;

}

public void insertLast(String index, String name, int points) {

if (first != null) {

SLLNode tmp = first;

while (tmp.succ != null)

tmp = tmp.succ;

SLLNode ins = new SLLNode(index, name, points, null);

tmp.succ = ins;

} else {

insertFirst(index, name, points);

}

}

@Override

public String toString() {

StringBuilder ret = new StringBuilder();

if (first != null) {

SLLNode tmp = first;

ret.append(tmp).append("\n");

while (tmp.succ != null) {

tmp = tmp.succ;

ret.append(tmp).append("\n");

}

} else

ret = new StringBuilder("NO ELEMENTS");

return ret.toString();

}

}

public class Students {

public static void removeStudent(SLL students) {

// todo: implement function

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = Integer.parseInt(scanner.nextLine());

SLL students = new SLL();

for (int i = 0; i < n; i++) {

String line = scanner.nextLine();

String[] parts = line.split("\\s+");

students.insertLast(parts[0], parts[1], Integer.parseInt(parts[2]));

}

removeStudent(students);

System.out.println(students.toString());

}

}

Modifikacii:

Izbrishi:

* Prviot so najmal broj na poeni ako postojat povekje so najmal broj na poeni
* Posledniot so najmal broj na poeni ako postojat povekje so najmal broj na poeni
* Site so min broj poeni ako ima povekje so najmal broj na poeni

6

161414 Luke 100

123242 Kate 67

151515 Sarah 89

123232 Maksut 67

131414 Emily 99

171717 Tom 67

Izlez:

Luke

Sarah

Emily

class Student{

String index;

String name;

int points;

SLLNode succ;

public Student(String index, String name, int points) {

this.index = index;

this.name = name;

this.points = points;

}

@Override

public String toString() {

return name;

}

}

public class Students {

public static void removeStudent(SLL<Student> students) {

// todo: implement function

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = Integer.parseInt(scanner.nextLine());

SLL<Student> students = new SLL<Student>();

Student st=null;

for (int i = 0; i < n; i++) {

String line = scanner.nextLine();

String[] parts = line.split("\\s+");

st=new Student(parts[0], parts[1], Integer.parseInt(parts[2]))

students.insertLast(st);

}

removeStudent(students);

System.out.println(students.toString());

}

}