Clase estudiante

```java
public class Student {

	private String lastName;
	private int id;
	private double average;

	public String getLastName() {
		return lastName;
	}
	public void setLastName(String lastName) {
		this.lastName = lastName;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public double getAverage() {
		return average;
	}
	public void setAverage(double average) {
		this.average = average;
	}

	public Student(String lastName, int id, double average) {
		super();
		this.lastName = lastName;
```

```
            this.id = id;
            this.average = average;
      }
}
```

Clase Search

```
import java.util.ArrayList;

public class Search {

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

      public static void search(ArrayList<Student> studentList,double average, int position ) {

        for(int i = 0;i < studentList.size();i = 100+i){
              if(i+100>=studentList.size()){
                    int h=studentList.size()%100;
                    boundedSearch(i,i+h+100,average,studentList);
              }else
                   boundedSearch(i,i+100,average,studentList);

        }
        if(indexList.size() >= position)
            System.out.println(studentList.get(indexList.get(position)).getLastName()
                      +" "+studentList.get(indexList.get(position)).getId());
      }
```

```
     public static void boundedSearch(int p1, int p2,double a,ArrayList<Student> studentList){
       for(int w = p1; w < p2-1; w++)
            if(studentList.get(w).getAverage()>=a)
                 indexList.add(w);
     }
}
```

clase next permutation

```
public class NextPermutation {

     public static <T extends Comparable> T[] nextPermutation(T[] array) {

         int lessPosition = array.length - 1;
         while (lessPosition > 0 &&
                array[lessPosition - 1].compareTo(array[lessPosition]) >= 0 )
          lessPosition--;

         if (lessPosition == 0){
             return array;
         }

         int higherPosition = array.length - 1;
         while (array[higherPosition].compareTo(array[lessPosition - 1]) <= 0 )
          higherPosition--;
```

```
            T temp = array[lessPosition - 1];
            array[lessPosition - 1] = array[higherPosition];
            array[higherPosition] = temp;

            higherPosition = array.length - 1;
            while (lessPosition < higherPosition) {
                temp = array[lessPosition];
                array[lessPosition] = array[higherPosition];
                array[higherPosition] = temp;
                lessPosition++;
                higherPosition--;
            }
            return array;
        }
}
```