Да се имплементира класа `List` во којашто ќе се чуваат информации за:

- броеви што се дел од листата (динамички алоцирана низа од цели броеви)
- бројот на броеви што се дел од листата

За класата да се дефинираат следните методи:

- конструктор (со аргументи), copy-конструктор, деструктор, оператор `=`
- `void pecati()` метод што ќе ги печати информациите за листата во форматот: `[број на елементи во листата]: x1 x2 .. xn sum: [сума] average: [просек]`
- `int sum()` метод што ја враќа сумата на елементите во листата
- `double average()` метод што ќе го враќа просекот на броевите во листата

Дополнително, креирајте класата `ListContainer`, во која што ќе се чуваат информации за:

- низа од листи (динамички алоцирана низа од објекти од класата `List`)
- број на елементи во низата од листи (цел број)
- број на обиди за додавање на листа во контејнерот (цел број првично поставен на нула)

За класата потребно е да ги дефинирате следните методи:

- конструктор (default), copy-конструктор, деструктор, оператор `=`
- `void pecati()` метод што ќе ги печати информациите за контејнерот во форматот: `List number: [реден број на листата] List info: [испечатени информации за листата со методот List::pecati()] \n sum: [сума] average: [просек]`
  - доколку контејнерот е празен се печати само порака `The list is empty`.
- `void addNewList(List l)` метод со којшто се додава листа во контејнерот
  - **Напомена**: една листа се додава во контејнерот ако и само ако има различна сума од сите листи што се веќе додадени во контејнерот
- `int sum()` метод што ја враќа сумата на сите елементи во сите листи

- `double average()` метод што го враќа просекот на сите елементи во сите листи во контејнерот

For example:

Input	Result
2 1 1 2 0 1 0	List number: 1 List info: 1: 1 sum: 1 average: 1 Sum: 1 Average: 1 Successful attempts: 1 Failed attempts: 1