Third element

Mary

The exercise contains a base that asks the user for strings and adds them to a list. The program stops reading when the user enters an empty string. The program then prints the first element of the list.

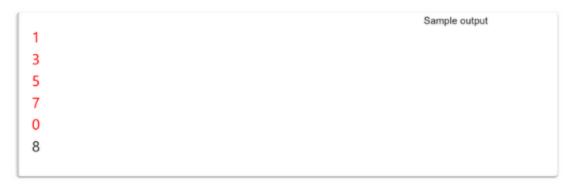
Your assignment is to modify the program so that instead of the first value, the third value on the list is printed. Remember that programmers start counting from zero! The program is allowed to malfunction if there are fewer than three entries on the list, so you don't need to prepare for such an event at all.

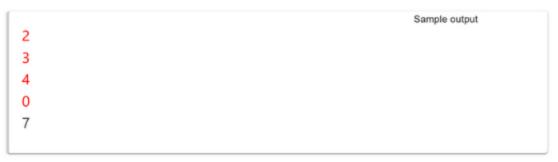
	Sample output
Tom	
Emma	
Alex	
Mary	
Alex	
	Sample output
Emma	
Alex	
Mary	
Mary	

Second plus third

In the exercise template there is a program that reads integers from the user and adds them to a list. This ends when the user enters 0. The program then prints the first value on the list.

Modify the program so that instead of the first value, the program prints the sum of the second and third numbers. The program is allowed to malfunction if there are fewer than three entries on the list, so you don't need to prepare for such an event at all.





IndexOutOfBounds Exception

A list is extremely useful for storing the values of variables for later use. That said, making mistakes is also relatively easy with a list.

There is a program that uses a list in the exercise template. Modify it so that its execution always produces the error IndexOutOfBounds. The user should not have to give any inputs to the program (e.g. write something on the keyboard)

You can also see a means for going through the values of a list — we will return to this topic a bit later.

List size

In the exercise template is a program that reads input from the user. Modify its working so that when the program quits reading, the program prints the number of values on the list.

Tom
Emma
Alex
Mary
In total: 4

Sample output

Sample output

lda Christopher Mark Sylvester Oscar

Irene Olivia Liam

In total: 11

NB! Be sure to use the size method of the list.

Last in list

In the exercise template there is a program that reads inputs from the user and adds them to a list. Reading is stopped once the user enters an empty string.

Your task is to modify the method to print the last read value after it stops reading. Print the value that was read last from the list. Use the method that tells the size of a list to help you.

Tom
Emma
Alex
Mary
Mary

Juno
Elizabeth
Mason
Irene
Olivia
Liam
Ida
Christopher
Mark
Sylvester
Oscar

First and last

In the exercise template there is a program that reads inputs from the user and adds them to a list. Reading is stopped once the user enters an empty string.

Modify the program to print both the first and the last values after the reading ends. You may suppose that at least two values are read into the list.

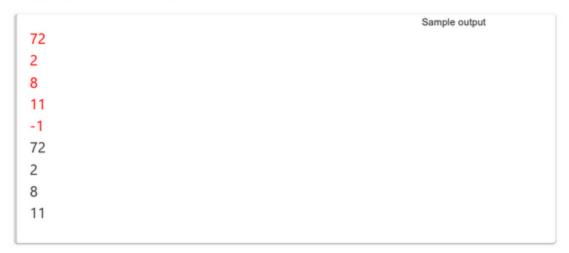
Tom
Emma
Alex
Mary
Tom
Mary

Juno
Elizabeth
Mason
Irene
Olivia
Liam
Ida
Christopher
Mark
Sylvester
Oscar
Juno
Oscar

Remember these numbers

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

Expand the functionality of the program so that after reading the numbers, it prints all the numbers received from the user. The number used to indicate stopping should not be printed.



Only these numbers

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

Expand the program to ask for a start and end indices once it has finished asking for numbers. After this the program shall prints all the numbers in the list that fall in the specified range (between the indices given by the user, inclusive). You may assume that the user gives indices that match some numbers in the list.

```
72
2
8
11
-1
From where? 1
To where? 2
2
```

```
72
2
8
11
-1
From where? 0
To where? 2
72
2
```

Greatest in list

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

Continue developing the program so that it finds the greatest number in the list and prints its value after reading all the numbers. The programming should work in the following manner.

```
72
2
8
93
11
-1
The greatest number: 93
```

You can use the source code below as an example. It is used to find the smallest number.

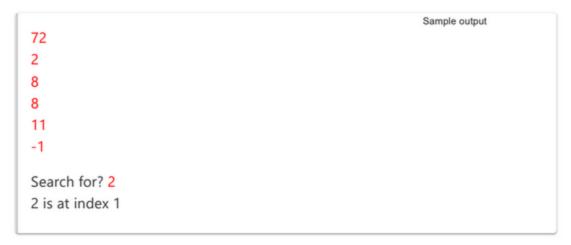
```
// assume we have a list that contains integers
int smallest = list.get(0);

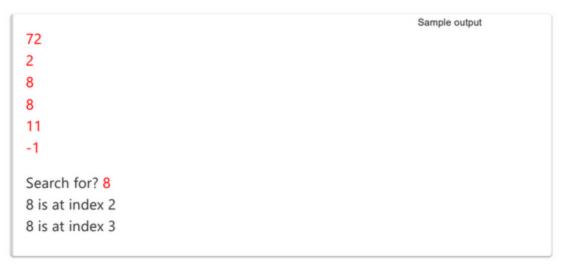
for(int i = 0; i < list.size(); i++) {
    int number = list.get(i);
    if (smallest > number) {
        smallest = number;
    }
}
System.out.println("The smallest number: " + smallest);
```

Index of

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

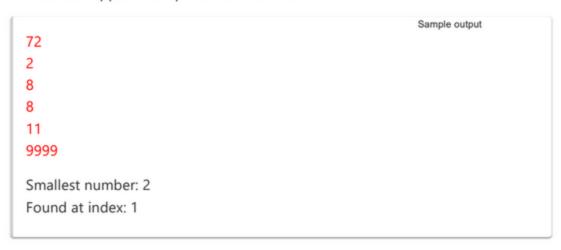
Expand the program by adding a functionality that asks the user for a number, and reports that number's index in the list. If the number is not found, the program should not print anything.

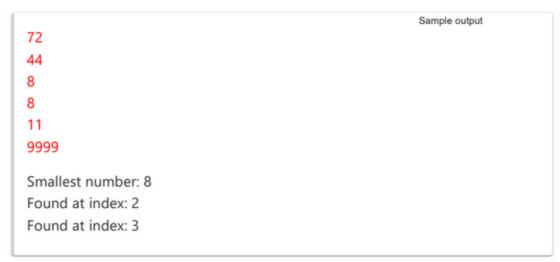




Index of smallest

Write a program that reads numbers from the user. When number 9999 is entered, the reading process stops. After this the program will print the smallest number in the list, and also the indices where that number is found. Notice: the smallest number can appear multiple times in the list.



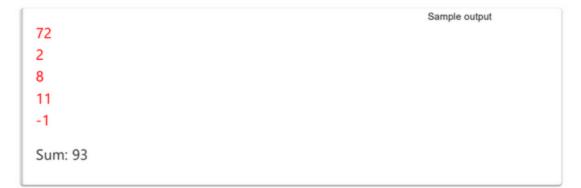


Hint: combine the programs you wrote for the exercises "Greatest number in the list" and "Index of the requested number". First find the smallest number, and then find the index of that number.

Sum of a list

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

Modify the program so that after reading the numbers it calculates and prints the sum of the numbers in the list.



Average of a list

The exercise template contains a base that reads numbers from the user and adds them to a list. Reading is stopped once the user enters the number -1.

When reading ends, calculate the average of the numbers in it, and then print that value.

72 2 8 11 -1 Average: 23.25

On the list?

In the exercise template there is a program that reads inputs from the user until an empty string is entered. Add the following functionality to it: after reading the inputs one more string is requested from the user. The program then tell whether that string was found in the list or not.

Sample output Tom Emma Alex Mary Search for? Mary Mary was found!

Tom Emma Alex

Mary

Search for? Logan

Logan was not found!

Sample output

Print in range

Create the method public static void printNumbersInRange(ArrayList<Integer> numbers, int lowerLimit, int upperLimit) in the exercise template. The method prints the numbers in the given list whose values are in the range [lowerLimit, upperLimit]. A few examples of using the method are supplied below.

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(3);
numbers.add(2);
numbers.add(6);
numbers.add(-1);
numbers.add(5);
numbers.add(1);

System.out.println("The numbers in the range [0, 5]");
printNumbersInRange(numbers, 0, 5);

System.out.println("The numbers in the range [3, 10]");
printNumbersInRange(numbers, 3, 10);
```

```
The numbers in the range [0, 5]
3
2
5
1
The numbers in the range [3, 10]
3
6
5
```

Sum

Create the method public static int sum(ArrayList<Integer>
numbers) in the exercise template. The method is to return the sum of the numbers in the parameter list.

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(3);
numbers.add(2);
numbers.add(6);
numbers.add(-1);
System.out.println(sum(numbers));
numbers.add(5);
numbers.add(1);
System.out.println(sum(numbers));
```

```
10
16
```

Remove last

Create the method public static void removeLast(ArrayList<String> strings) in the exercise template. The method should remove the last value in the list it receives as a parameter. If the list is empty, the method does nothing.

```
ArrayList<String> strings = new ArrayList<>();

strings.add("First");
strings.add("Second");
strings.add("Third");

System.out.println(strings);

removeLast(strings);
removeLast(strings);
System.out.println(strings);
```

```
Sample output
[First, Second, Third]
[First]
```