

04_11.Counting	2
04_12.CountingToHundred	3
04_13.FromWhereToWhere	4
04_14.SumOfASequence	6
04_15.SumOfASequenceTheSequel	7
04_16.Factorial	8

Counting

Write a program that reads an integer from the user. Next, the program prints numbers from 0 to the number given by the user. You can assume that the user always gives a positive number. Below are some examples of the expected functionality.

Sample output

```
4
0
1
2
3
4
```

Sample output

```
1
0
1
```

Programming exercise:

Counting to hundred

Write a program, which reads an integer from the user. Then the program prints numbers from that number to 100. You can assume that the user always gives a number less than 100. Below are some examples of the expected functionality.

Sample output

99
99
100

Sample output

-4
-4
-3
-2
-1
0
1
2
... (many numbers in between) ...
98
99
100

From where to where? (2 parts)

Where to

Write a program which prints the integers from 1 to a number given by the user.

Sample output

Where to? 3

1
2
3

Sample output

Where to? 5

1
2
3
4
5

hint the number read from the user is now the upper limit of the condition.
Remember that in Java `a <= b` means *a is smaller or equal to b*.

Where from

Ask the user for the starting point as well.

Sample output

Where to? 8

Where from? 5

5
6
7
8

If the upper limit is larger than the starting point, nothing is printed:

Sample output

Where to? 12

Where from? 16

NB remember that the lower and upper limits can be negative!

Sum of a sequence

Implement a program, which calculates the sum $1+2+3+\dots+n$ where n is given as user input.

Sample output:

Last number? 3

The sum is 6

Sample output

The previous example calculated $1 + 2 + 3 = 6$

Last number? 7

The sum is 28

Sample output

And this one calculated $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$

Sum of a sequence - the sequel

Implement a program which calculates the sum of a closed interval, and prints it. Expect the user to write the smaller number first and then the larger number.

You can base your solution to this exercise to the solution of last exercise — add the functionality for the user to enter the starting point as well.

Sample output:

Sample output

First number? 3

Last number? 5

The sum is 12

The above example internally calculated $3 + 4 + 5 = 12$

Sample output

First number? 2

Last number? 8

The sum is: 35

And now the internal calculation was $2 + 3 + 4 + 5 + 6 + 7 + 8 = 35$

Programming exercise:

Factorial

Implement a program which calculates the factorial of a number given by the user.

Factorial of n , denoted $n!$, is calculated as $1 * 2 * 3 * \dots * n$. For example, the factorial of 4 is 24 or $4! = 1 * 2 * 3 * 4 = 24$. Additionally, it has been specified that the factorial of 0 is 1, so $0! = 1$.

Sample output:

Sample output

Give a number: 3

Factorial: 6

The internal calculation here was $1 * 2 * 3 = 6$

Sample output

Give a number: 10

Factorial: 3628800

And now the internal calculation was $1 * 2 * 3 * \dots * 8 * 9 * 10 = 3628800$

Additional info: Factorials are used especially in probability calculus when examining different possible orders of a set. For example, a group of five people can form $5!$ different lines, and a pack of 52 cards can be in $52!$ different orders. Factorial can also be used to calculate [combinations](#); For example it is possible to deal $52! / (5! * (52-5)!)$ different hands from a 52 card pack, and you can form $40! / (7! * (40-7)!)$ different 7 number lottery lines from 40 numbers.