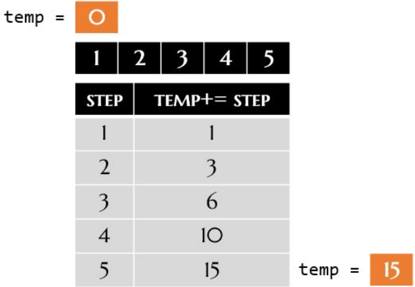
Programming I - Laboratory lesson 6,7

Sead Jahić
Teaching Assistant, Information Technologies

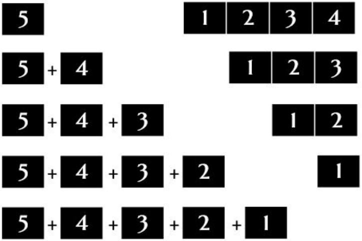
University of Primorska
Faculty of Mathematics, Natural Sciences and Information Technologies
(UP FAMNIT)

November 28th, 2022.

Suppose that we need to sum first n elements. There are several solutions, and some of them are:



STEP	TEMP+= STEP
1	1
2	3
3	6
4	10
5	15



You can check you answer by using $\frac{n(n+1)}{2}$ (formula for the sum of the first n numbers).

Exercise

As in previous example try to find sum of:

- *sum of the first n odd numbers*
- *sum of the first n even numbers*

using recursion.

Exercise

Find n^{th} power of number a, where $a > 0; a \in \mathbb{N}$.

Explanation:

1. we find $(n-1)^{th}$ power of a and multiply the result by a,
2. to calculate $(n-1)^{th}$ power of a, we calculate $(n-2)^{th}$ power of a and multiply the result by $a \ldots$.

Exercise

Factorial of number $n \geq 0$.

Exercise

Fibonacci sequence - recursion

More about Fibonacci sequence you can find on web page: `https://en.wikipedia.org/wiki/Fibonacci_number`. The Fibonacci Sequence is the series of numbers:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

The next number is found by adding up the two numbers before it. The sequence F_n of Fibonacci numbers is defined by the recurrence relation:

$$F_n = F_{n-1} + F_{n-2},$$

$$F_0 = 0, F_1 = 1.$$

- Arithmetic progression more on site:`https://en.wikipedia.org/wiki/Arithmetic_progression`

In mathematics, an arithmetic progression (AP) or arithmetic sequence is a sequence of numbers such that the difference between the consecutive terms is constant.
Formula:

$$a_n = a_1 + (n-1) \cdot d,$$

but after calculating, we can get formula:

$$a_n = a_{n-1} + d,$$

so, in this order we can use recursion to implement program for Arithmetic progression. More explanation on next frame.

Let say that we want to have 5−th element of progression: $2, 5, 8, 11, 14, 17, 20, \ldots$. As we can see it is element 14, first element $a_1 = 2$ and difference $d = 3$. So, recursion should be implemented by using for-loop (for $i = 1$ we should return a_1). We will use sum (have to be declared before for loop) because every time we add difference to previous value of element.

$i = 2 \Rightarrow$ PE$= 2$ (position $n - 1$) + d ($= 3$) equals 5
$i = 3 \Rightarrow$ PE$= 5$ (position $n - 1$) + d ($= 3$) equals 8
$i = 4 \Rightarrow$ PE$= 8$ (position $n - 1$) + d ($= 3$) equals 11
$i = 5 \Rightarrow$ PE$= 11$ (position $n - 1$) + d ($= 3$) equals 14

And geometric progression is given as

$$q_n = q_{n-1} \cdot r,$$

where q_n is $n-$th element of GP, and $r = \dfrac{q_n}{q_{n-1}}$, as it is obviously.

Exercise

Greatest common divisor

To find GCD (greatest common divisor) we will use Euclidean algorithm. Best way to see how we can implement this program, with recursion, is on some example. Let say that we want to find GCD (1071, 462).

Step k	Equation	Quotient and remainder
0	$1071 = q_0 \cdot 462 + r_0$	$q_0 = 2$ and $r_0 = 147$
1	$462 = q_1 \cdot 147 + r_1$	$q_1 = 3$ and $r_1 = 21$
2	$147 = q_2 \cdot 21 + r_2$	$q_2 = 2$ and $r_2 = 0$.

For illustration, the GCD(1071, 462) is calculated from the equivalent GCD(462, 1071 mod 462) = GCD(462, 147). The latter GCD is calculated from the GCD(147, 462 mod 147) = GCD(147, 21), which in turn is calculated from the GCD(21, 147 mod 21) = GCD(21, 0) = 21.

Or better illustration:

GCD(1071, 462)

GCD(462, 1071 mod 462)

GCD(147, 462 mod 147)

GCD(21, 147 mod 21)= GCD(21, 0) = 21