

Algorithms

Lab 04 (8%)

Topics: Recursion, Backtracking

Problem 01 (0.25%)

(Simplest illustration of recursion)

Write a C++ program which reads arbitrary amount of integer numbers from standard input and prints them to standard output in reversed order. Do not use loops and containers. Use recursive function `readAndPrintInReversedOrder` instead. Explain disadvantages of this recursive approach.

Problem 02 (0.25%)

(Factorial)

Write a C++ program which computes factorial of entered non-negative integer numbers `n` (Ctrl-Z – end of input). Use recursive function `int64_t factorial(int n)`. Explain disadvantages of this recursive approach.

Problem 03 (1%)

(Greatest Common Divisor)

Write a C++ program which computes greatest common divisors of entered pairs of integer numbers (Ctrl-Z – end of input). Create functions:

- `int gcd(int a, int b)`: checks arguments and throws `std::invalid_argument` exception if `a == 0` and `b == 0`, otherwise it calls recursive function `int gcdAux(int a, int b)`
- `int gcdAux(int a, int b)`: recursive function to compute greatest common divisor of numbers `a` and `b`.

Problem 04 (1%)

(Hanoi Towers)

Write a C++ program which solves recursively Hanoi Towers puzzle for entered non-negative integer `n`. Program has to write solution as a sequence of lines `<number of source tower> → <number of destination tower>`.

Sample input:

3

Sample output:

1->3

1->2

3->2

1->3

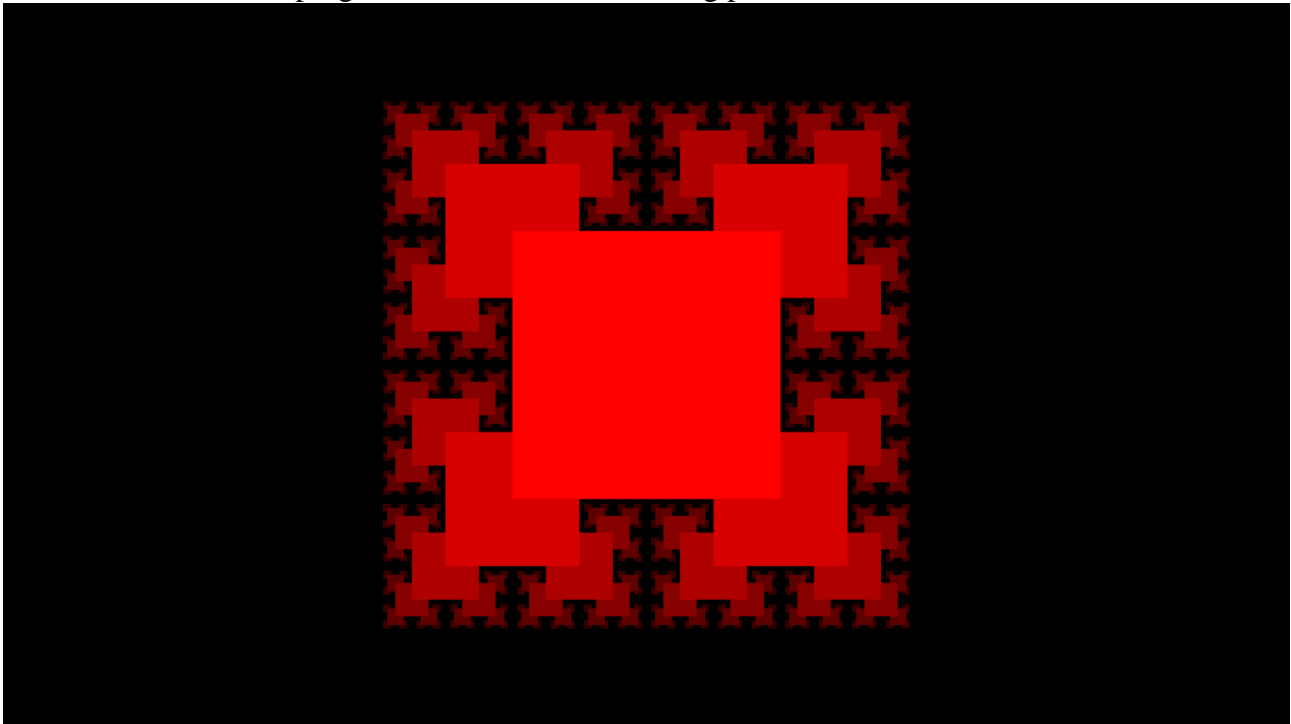
2->1

2->3

1->3

Problem 05 (1%)
(Simplest Fractal)

Write a recursive Java program which creates following picture:



Problem 06 (0.5%)
(Example of bad usage of recursion)

Write a C++ program which computes nth element of Fibonacci sequence. Use direct recursive implementation of function:

$F(n) = 1$, if $n = 0$, $n = 1$;
otherwise
 $F(n) = F(n - 1) + F(n + 2)$.

Explain why this function is so inefficient. Solve the same problem without recursion.

Problem 07 (2%)
(Backtracking: recursive approach)

Write a C++ program which solves famous N Queens puzzle for entered natural n – size of chessboard. Program has to print all configuration of n queens where queens do not attack each other. Use recursive function to solve this puzzle.

Problem 08 (2%)
(Backtracking: iterative implementation)

You have to solve previous problem using iterative function instead of recursive one. (Hint: use class stack from standard library)