Programming Prerequisites

Philip Bille   Inge Li Gørtz

These exercises aim to determine if you are ready to study algorithms or have to refresh your programming skills beforehand. Read the exercises carefully and consider how you would solve them.

- If you can immediately see what you should do to solve the exercises and are confident that you can easily solve them, then you are already ready for the programming challenges in algorithmics. You do not need to implement the solutions.
- If you are unsure how to solve one or a few of the exercises, you should take some time to implement those exercises to ensure you are ready for this course.
- If you are unsure how to solve several of the exercises, you probably need to refresh your programming skills. Find a concise book on programming focusing on the basics, such as "Introduction to Programming in Python," by Sedgewick and Wayne.

Exercises

We recommend using a simple text editor (not an IDE like Eclipse or similar) and compiling and executing directly from a console window, especially for exercises using standard input and output. Avoid the use of built-in and external libraries/packages in your solutions.

1 Calculation Write a function `sub` that, given two integers a and b, returns $a - b$.

2 Cut-off Write a function `cut-off` that, given an array A of integers and an integer k, changes the array such that all integers in A with values $> k$ are changed to k.

3 Input and Output Solve the following exercises.

3.1 Write a program `print` that outputs `Hey World` to the console (ie. to `stdout`).

3.2 Extend the program to take a string s as command line argument and output `Hey` followed by s.

3.3 Write a program `interaction` that reads a string s from the console (`stdin`) and outputs `Hey` followed by s.

4 Sum Write a function `sum` that takes a positive integer x as argument and returns the sum $\sum_{i=1}^{x} i = 1+2+3+\cdots+x$.

5 Order Check Write a function `inorder` that takes three integers a, b, and c as arguments and returns true if the values are in strictly increasing or strictly decreasing order (i.e., $a < b < c$ or $a > b > c$) and otherwise false.

6 Exchange of Elements

6.1 Write a function `swap` that given an array A and two indexes i and j exchanges the elements on index i and j in A. Output the elements of A before and after the exchange.

6.2 Write a function `reverse` that given an array A reverses the order of the elements in A.

7 Distance Write a function `dist` that, given two numbers x and y, returns the (euclidean) distance between the origin $(0,0)$ and (x,y) in the plane.

8 Three Sort Write a function `sort` that, given three integers a, b, and c, outputs the integers in order.

9 Factorial Write a function `fact` that, given an integer x, computes $x! = x \cdot (x-1) \cdots 1$. Your solution must be recursive, i.e., `fact` must compute $x!$ by calling itself.

10 Heads or Tails Write a function `flip` that, given an integer n and a probability p, simulates n coin throws with a probability of p for flipping heads. Output the sequence of outcomes.

11 Binary Integers Write a function `bin` that, given a positive integer x, outputs the binary representation of x.

12 Matrix Multiplication Write a function `matrixMul` that, given two $n \times n$ matrices A and B as 2-dimensional arrays, computes the matrix product AB.

13 Longest Plateau Write a function `plateau` that, given an array A of integers, finds the longest consecutive sequence of equal values where the value just before and the value just after are smaller.