Introduction to Computer Science
Tutorial 6: algorithmic problems, input/output, argc

1 Design and implement a program in C, that receives as input from standard input, a positive integer number n, and prints out the list of the first n natural numbers, indicating which are prime numbers, and which are composite numbers. The prime numbers must be highlighted in green, and the composite numbers in red. Number 1 should be left with the default color.

2 Design and implement a program in C, that receives as input from standard input, a positive integer number n, and prints out the number in base 2, i.e., as a sequence of 0s and 1s.

3 Design and implement a program in C, that receives as input from standard input, a sequence of 0s and 1s (a sequence of characters) representing a binary number, and prints out the corresponding number in base 10.

4 Design and implement a program in C, that receives as input from standard input, a positive integer number n (the number of inputs that follow), followed by n integer numbers x1, …, xn, computes the sum x1 + x2 + … + xn, and outputs it on standard output.

5 Design and implement a program in C, that receives as input from standard input, a positive integer number n (the number of inputs that follow), followed by n integer numbers x1, …, xn, and computes the median of x1, …, xn. The result should be outputted on standard output.

6 Design and implement a program in C, that receives as input from standard input, a string of characters, and outputs each character with its corresponding frequency (number of times it appears in the input string). Only characters with positive frequencies should be printed out. The characters with a maximum frequency must be highlighted in green. The characters with a minimum (positive) frequency should be highlighted in red.

7 Design and implement a program in C, that receives as input from standard input, a positive number n in the interval [0,127], and a string of characters, and generates (and prints out) a new string where every character in the input string is replaced by the corresponding character "left shifted" n times (e.g., if n is 3, the character with ASCII code 53 will be replaced by the character with ASCII code 50). Use modulo if the encoding goes below 0, to always get a code in the range [0,127].

8 Write a program that reads a positive integer from standard input and checks if it is a prime number. If the program is called with parameters, it must print an error message (in red) indicating that the input is expected from standard input, and terminate.