Imperial College
London

Coursework 1 (Assessed):
The Fundamentals of C

Lecturer: Dr. Paul A. Bilokon

Date set: 2024.01.25
Date due: 2024.02.08

Abstract

This coursework explores the foundations of programming in general and the programming language C in particular. It covers types, operators, and expressions; control flow; functions and program structure; pointers and arrays; structures.

1. Write a program which reads in two characters and outputs them in reverse order.

2. How many times will the loops defined by the following `for` loops be executed?

   (a) `for (i=10; i<=10; i++);`

   (b) `for (i=10; i<10; i++);`

   (c) `for (i=10; i>=1; i-=4);`

   (d) `for (i=10; i==1; i-=4);`

3. Write a program that reads numbers until a negative number is read and prints out the number of values read, the largest value, the smallest value, and the range.

4. Write a program which reads in 200 characters and outputs them in reverse order. Do not use arrays. Hint: use recursion.

5. When his teacher asked Sleepy to simplify the fraction 26/65, he simply cancelled the digit 6 both from the numerator and denominator. To the teacher's amazement, Sleepy's cancellation technique produced the correct result:

   $$\frac{26}{65} = \frac{2}{5}.$$

   Write a program to determine all the fractions with two-digit numerators and denominators for which Sleepy's technique gives the right answer. (Split your program into functions and divide them between files in an ordered way.)

6. Write a program that allows two users to play noughts and crosses. The program should ask for moves alternately from player X and player O. The program displays the game positions as

```
123
456
789
```

The players enter their moves by providing the position number. After each move the program displays the changed board. A sample board configuration is

```
XXO
OX6
7X0
```

(Split your program into functions and divide them between files in an ordered way.)

7. The sieve of Eratosthenes is a method for generating lists of primes. Start with a list of integers, say

$$2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19.$$

Mark every multiple of two (excluding two itself, so $4, 6, 8, \ldots$):

$$2, 3, \mathbf{4}, 5, \mathbf{6}, 7, \mathbf{8}, 9, \mathbf{10}, 11, \mathbf{12}, 13, \mathbf{14}, 15, \mathbf{16}, 17, \mathbf{18}, 19.$$

Then go to the next unmarked number, which is three, and mark all its multiples ($6, 9, 12, \ldots$):

$$2, 3, \mathbf{4}, 5, \mathbf{6}, 7, \mathbf{8}, \mathbf{9}, \mathbf{10}, 11, \mathbf{12}, 13, \mathbf{14}, \mathbf{15}, \mathbf{16}, 17, \mathbf{18}, 19.$$

Continue this process up to and including the greatest integer $\leq \sqrt{19}$. The numbers that remain unmarked are prime.

(a) Implement the sieve of Eratosthenes in C. Use your implementation to find all primes less than or equal to a given natural number N.

(b) The algorithm that we have described could be optimized by excluding all even numbers from the list. This optimization enables us to halve the amount of required memory. Implement an alternative version of the sieve making use of this optimization.

(Split your program into functions and divide them between files in an ordered way.)

8. Write functions that read, add, subtract, multiply, and evaluate polynomials of the form

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

Such polynomials can be represented by a linked list in which each node has three fields: coefficient a_i, exponent i, and pointer to the next node. For evaluating a polynomial, use Horner's method:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = (\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0.$$

(Split your program into functions and divide them between files in an ordered way.)