Applicants' programming exercises 2018

Department of Computer Science, UiT The Arctic University of Norway
Programming exercises for applicants to the Computer Science Master studies in Tromsø

These exercises must be solved using the *Python* or *C* programming languages.

The submitted code should be well documented through comments in the code or by other means. The code should be self contained: the code should not depend on code or tools not included in a standard distribution of the programming language you have chosen. The documentation should also include information about how to compile (for `C` code) and execute the submitted code. We should be able to test the submitted code on a standard installation of `C` or `Python`.

These exercises are used to screen applicants for their programming knowledge and background. These exercises are not meant to illustrate for the applicants the level of programming knowledge and experience we expect from them for our Master studies in Computer Science in Tromsø. As a master student in Computer Science in Tromsø you are expected to solve programming tasks that are more complicated than shown here.

Assignment: Integer to roman numeral, and back again

Roman numerals use the symbols `I` (1), `V` (5), `X` (10), `L` (50), `C` (100), `D` (500) and `M` (1000). The symbols are combined and added to create numeric values. As an example, the number 32 can be written `XXXII`. Subtraction notation is used to to avoid confusion between, for instance, `IIII` and `III`. This takes the form of a smaller value in front of a larger, such as `IX`, which represents 9, or `IV`, which represents 4.

See the appendix *Roman numerals* for a more complete description that covers the variations necessary for this assignment

(a) Implement a *C* or *Python* function `int_to_roman()` that converts an integer to a roman numeral. The input is guaranteed to be within the range of 1 to 3999 (inclusive).

```python
def int_to_roman(val):
    # Python function implementation here
```

```c
char *int_to_roman(int val) {
    /* C function implementation here */
}
```

The solution should support the subtractive notation with the values `IV` (4), `IX` (9), `XL` (40), `XC` (90), `CD` (400), and `CM` (900). For example, if the input to the function is `1994`, the output should be `"MCMXCIV"` (since `M`=1000, `CM`=900, `XC`=90 and `IV`=4).

(b) Implement a *C* or *Python* function `roman_to_int()` that takes the Roman numerals produced by the function `int_to_roman()` in assignment (a) and convert them back to integers. The input is guaranteed to be within the same range as in (a).

```python
def roman_to_int(numeral):
    # Python function implementation here
```

```c
int roman_to_int(char *numeral) {
    /* C function implementation here */
}
```

(c) **Optional:** Create a program that uses the functions from (a) and (b) to write out all the numbers from 1 to 3999 as Roman numerals, and then convert them back to, and print them out, as integers.

Appendix

Roman numerals

The following description is extracted from the Wikipedia article on Roman Numerals[1]. This copy only includes the relevant parts for this assignment. Please refer to the Wikipedia entry if you are interested in more details.

The numeric system represented by Roman numerals originated in ancient Rome and remained the usual way of writing numbers throughout Europe well into the Late Middle Ages. Numbers in this system are represented by combinations of letters from the Latin alphabet. Roman numerals, as used today, are based on seven symbols:

Symbol:	`I`	`V`	`X`	`L`	`C`	`D`	`M`
Value:	1	5	10	50	100	500	1000

The original pattern for Roman numerals used the symbols `I`, `V`, and `X` (1, 5, and 10) as simple tally marks. Each marker for 1 (`I`) added a unit value up to 5 (`V`), and was then added to `V` to make the numbers from 6 to 9:

`I`, `II`, `III`, `IIII`, `V`, `VI`, `VII`, `VIII`, `VIIII`, `X`

The numerals for 4 (`IIII`) and 9 (`VIIII`) proved problematic (among other things, they are easily confused with `III` and `VIII`), and are generally replaced with `IV` (one less than 5) and `IX` (one less than 10). This feature of Roman numerals is called subtractive notation. The numbers from 1 to 10 (including subtractive notation for 4 and 9) are expressed in Roman numerals as follows:

`I`, `II`, `III`, `IV`, `V`, `VI`, `VII`, `VIII`, `IX`, `X`

The system being basically decimal, tens and hundreds follow the same pattern. Thus 10 to 100 (counting in tens, with `X` taking the place of `I`, `L` taking the place of `V` and `C` taking the place of `X`):

`X`, `XX`, `XXX`, `XL`, `L`, `LX`, `LXX`, `LXXX`, `XC`, `C`

Note that 40 (`XL`) and 90 (`XC`) follow the same subtractive pattern as 4 and 9.

Similarly, 100 to 1000 (counting in hundreds):

`C`, `CC`, `CCC`, `CD`, `D`, `DC`, `DCC`, `DCCC`, `CM`, `M`

Again, 400 (`CD`) and 900 (`CM`) follow the standard subtractive pattern.

Many numbers include hundreds, units and tens. The Roman numeral system being basically decimal, each *place* is added in descending sequence from left to right, as with Arabic numerals. For example:

- 39 = "*Thirty nine*" (`XXX`+`IX`) = `XXXIX`
- 246 = "*Two hundred and forty six*" (`CC`+`XL`+`VI`) = `CCXLVI`

As each *place* has its own notation there is no need for place keeping zeros, so *missing places* are ignored, as in Latin (and English) speech, thus:

- 207 = "*Two hundred and seven*" (`CC`+`VII`) = `CCVII`
- 1066 = "*A thousand and sixty six*" (`M`+`LX`+`VI`) = `MLXVI`

Roman numerals for large numbers are nowadays seen mainly in the form of year numbers (other uses are detailed later in this article), as in these examples:

- 1776 (`M`+`DCC`+`LXX`+`VI`) = `MDCCLXXVI` (the date written on the book held by the *Statue of Liberty*)
- 1954 (`M`+`CM`+`L`+`IV`) = `MCMLIV` (as in the trailer for the movie *The Last Time I Saw Paris*)
- 1990 (`M`+`CM`+`XC`) = `MCMXC` (used as the title of musical project Enigma's debut album *MCMXC a.D.*)
- 2014 (`MM`+`X`+`IV`) = `MMXIV` (the year of the games of the *XXII Olympic Winter Games* in Sochi)

The Wikipedia entry contains more information about the number system, but we will focus on the above description.

[1] `https://en.wikipedia.org/wiki/Roman_numerals`