University Göttingen
GWDG/AG-C
Julian Kunkel

Exercise 1 / 2023.04.17
Practical course on High-Performance Computing / SoSe 2022/23
25 Minutes Total

Welcome

This exercise is about familiarizing yourself with the format and to get to know the other peers. Typically, you will get one exercise sheets with multiple tasks.

Contents

Task 1: Understand the Practical Structure (3 min)

A tutorial equips you with additional knowledge and tools and teaches you how to use them effectively to complete the exercises. When teaching tools, the lecturer may give you an overview and then will walk through an interactive guide to overcome the typical obstacles faced by newcomers.

As part of the practical in the exercise, we will employ *active learning* in the form of group work. In the exercise, there is typically time for you to play with the tools and perform some tasks. You may also use this time to discuss issues with your colleagues. By providing further details, the lecture will be augmented. Part of the sheets will be composed by *links* to aid your independent study of specific topics. Some group work is theoretical and used to augment the lectures further, for instance, to obtain feedback from your peers and reflect on your skills.

There might be optional tasks that you can do instead of another task.

Remember:

- The lecture/tutorial is given in the main room, keep this chat quiet, except during the plenary and questions for a lecture. Stay in the main room at all times to listen to the lecturer's instructions.
- You discuss in your study group in breakout rooms using screen sharing and mic
  - Unresolvable problems are posted in the support room chat
- In the plenary we share achievements, challenges – you may ask anything!
  - Please volunteer to share results such that your peers can learn

Organization of the tasks The worksheets typically contain semantically the following sections for a tutorial/exercise:

- Post a hello on the break out room chat!

- Short introductory paragraph describing the topic/task. You are supposed to read it.
- Steps, which is a set of tasks you should perform.
- Knowledge/Background providing you with the reference knowledge needed to solve the task. This is your reference you can read during the tutorial.
- Hints, providing some additional tips for helping you to obtain the goal.
- Literature, a section containing references for further reading (beyond what we offer during the course).

Typically, you will find at each task a set of steps for you to process, which looks like this:

Steps

1. First you do this...
2. Then do that...

The time you should spend on a task may be indicated to help you with the time management.

Make sure you listen to the main room with the lecturer at all times.

Task 2: Introduction in the Study Group (7 min)

As part of this task, everyone introduces themself to the peers in the study group and tries to utilize the basics of BBB.

Steps

1. Introduce yourself briefly to your study group (name and 1-2 sentences regarding your background)
2. Have at least one person of the group try to share the screen - this will be a useful tool to discuss issues
3. If time permits: everyone can state their experience with group work/study works

Hints

- Allow everyone to participate
- Be kind to each other
- Take turns to present your result

Further Reading

- https://learningcenter.unc.edu/tips-and-tools/study-partners (Section: Tips for effective studying in groups)

Task 3: Team-based Problem Solving (15 min)

To break the ice on collaborating with the other members of the Study Group, we propose the following friendly competition between groups: a round of **Code Golf**. Given the problem statement and a starting Python program, you should collectively work on reducing the size of that program (total number of character), while preserving the core feature (still solves the problem, and can be used as shown below). The Study Group with the smallest working program wins!

Problem statement: Write a function taking as argument an integer between 1 and 2000, and returning a string of the equivalent Roman Numeral.

Starting program:

roman.py

```python
def integer2roman(n):

    def doOneDigit(digit, caseNine, caseFive, caseOne):
        if digit == 9:
            return caseNine
        elif digit >= 5:
            return caseFive + caseOne * (digit - 5)
        elif digit == 4:
            return caseOne + caseFive
        else:
            return caseOne * digit

    romanNumeral = ""
    romanNumeral = romanNumeral + ("M" * (n//1000))

    digitHundred = (n // 100) % 10
    romanNumeral = romanNumeral +  doOneDigit(digitHundred, 'CM', 'D', 'C')

    digitTens = (n // 10) % 10
    romanNumeral = romanNumeral +  doOneDigit(digitTens, 'XC', 'L', 'X')

    digitOnes = n % 10
    romanNumeral = romanNumeral +  doOneDigit(digitOnes, 'IX', 'V', 'I')

    return romanNumeral
```

Example usage:

```python
for i in range(1,2001):
    romnum = integer2roman(i)
    print(romnum)
```

Steps

1. One member of the study group shares their screen with a python interpreter or notebook, and will update/run the program with the suggestions of the team
2. Iteratively replace parts of the program with shorter alternatives
3. After **10 minutes**, one of you will share your final program in the Shared Note section of the main room, and announce the length of your program in the main chat.

Hints

- Start by discussing general strategies
- Don't neglect the corner cases: your final program will have to return the same values as the starting program for the whole problem domain
- The name of the function should stay the same - to allow for the final verification
- Don't forget to write your names in the shared notes, along with your program. You may optionally come up with a collective name for your Study Group.

Optional Task 4: Interact on the support room (2 min)

This is a difficult **additional** task that will support your understanding in the topic.

This is a very ordinary extra Task... To ensure you know where to get help when stuck, interact on the support room.

Steps

1. Post a hello on the support room chat!