

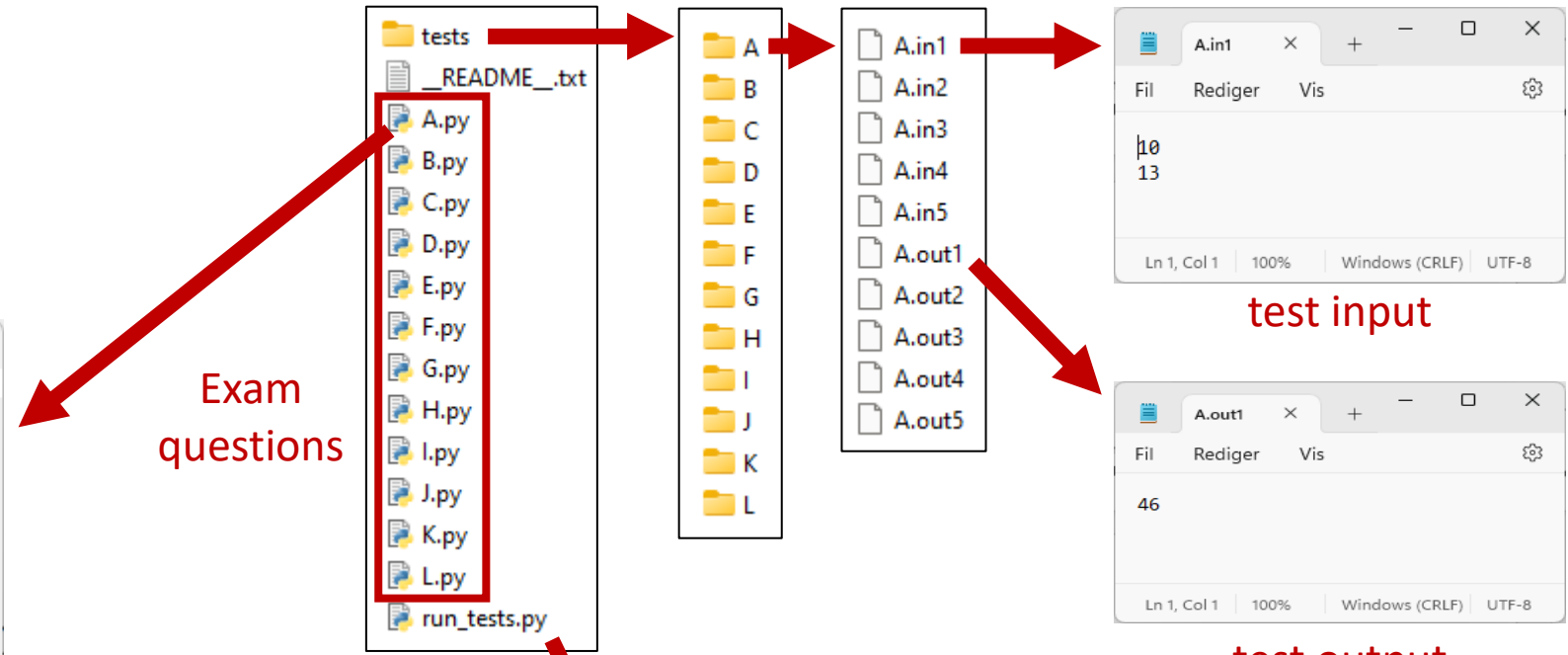
Introduction to Programming with Scientific Applications

- Missing handins, deadline 19 May 2023
- Final project, deadline 31 May 2023
- Course evaluation
- Exam, 17 June 2023
- AOB

Exam

- **5 hours, written exam, with aids, including PC and internet**
- **Communication with others about the exam is not permitted during the exam**
- **ITX Flex must be enabled**
- **AI assistants like ChatGPT and GitHub Copilot are not allowed**
- Reexam in August
- Grade is an *overall assessment* of the implementation project and the exam
 - The result of the final exam must meet the minimum requirements for acceptance to be able to pass the course
 - The final exam contributes 80% to the final grade
- eksamen.au.dk
 - Download .zip + add missing code + upload .zip
- Questions? – post them on Brightspace

Content of .zip file



Exam questions

test input

test output

```
A.py - C:\Users\au121\Desktop\ipsa22\ipsa22exam\A.py (3.11.3)
File Edit Format Run Options Window Help
...
INTERVAL SUM

Your task is to write a function interval_sum(i, j), that returns
the sum i + (i + 1) + ... + j. Eg. for i = 10 and j = 13 the sum
returned should be 10 + 11 + 12 + 13 = 46.

Input: Two lines, containing integers i and j, respectively.
It is guaranteed that 1 <= i <= j <= 100.

Output: The sum i + (i + 1) + ... + j.

Example:
Input: 10
      13
Output: 46

Note: The below code already handles the input and output.
...
def interval_sum(i, j):
    # insert code
    pass

i = int(input())
j = int(input())
print(interval_sum(i, j))
Ln: 1 Col: 0
```

```
Command Prompt
(c) Microsoft Corporation. All rights reserved.
C:\Users\au121>cd Desktop\ipsa22\ipsa22exam
C:\Users\au121\Desktop\ipsa22\ipsa22exam>python run_tests.py A --verbose=-1

Tests passed:
A      0/5
-----
Total  0/5
=====

C:\Users\au121\Desktop\ipsa22\ipsa22exam>python run_tests.py A --verbose=3 --abort
A.py tests/A\A.in1 [failed]
Input
> 10
> 13
Correct output
> 46
Received output
*> None
C:\Users\au121\Desktop\ipsa22\ipsa22exam>
```

* = line with wrong output

- run_tests.py arguments
- A B C ...
exercises to evaluate
default are all
 - --abort
stop on first error
 - --verbose=value
amount of output

Evaluation of code

- Each problem will be assigned a **weight**
- There will be problems of **varying difficulty**
- Code will be evaluated on **known test cases** and **unknown test cases**
- In general, **automatic scoring**, in some exceptional cases manual
- Googling / stack overflow / Python documentation etc. **is allowed**, but put a **comment if you copied code from internet** to avoid plagiarism

Don't expect partial scores
for this solution

```
def interval_sum(i, j):  
    if i == 10 and j == 13:  
        return 46
```

AOB ?