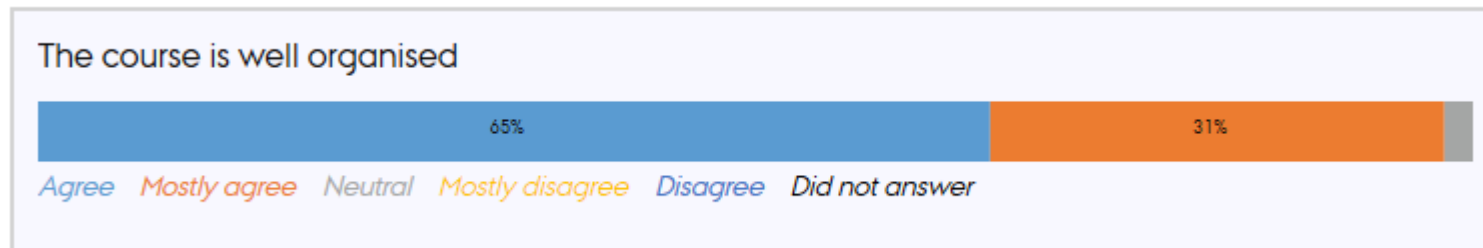
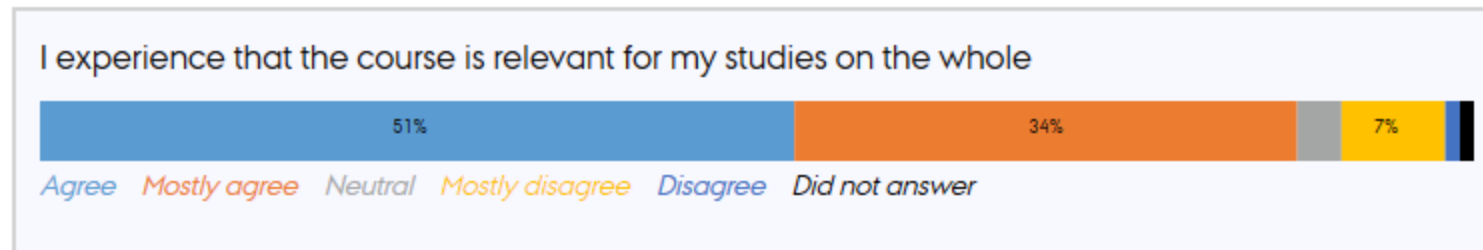
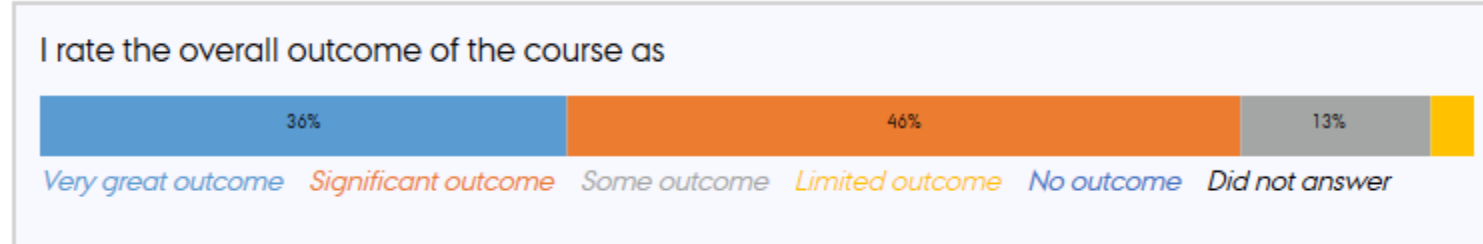


Introduction to Programming with Scientific Applications

- Missing handins – deadline: ASAP, but latest May 31st
- Final project – deadline: May 31st
 - June 1st the exam office is informed who passed the mandatory course assignments
- Course evaluation
- Exam
- AOB

Course evaluation



- "Not that relevant for my math studies... but will be super relevant when entering the real world"
- Videos sometimes delayed

Your background

How important was it to you that the course was a study café ?



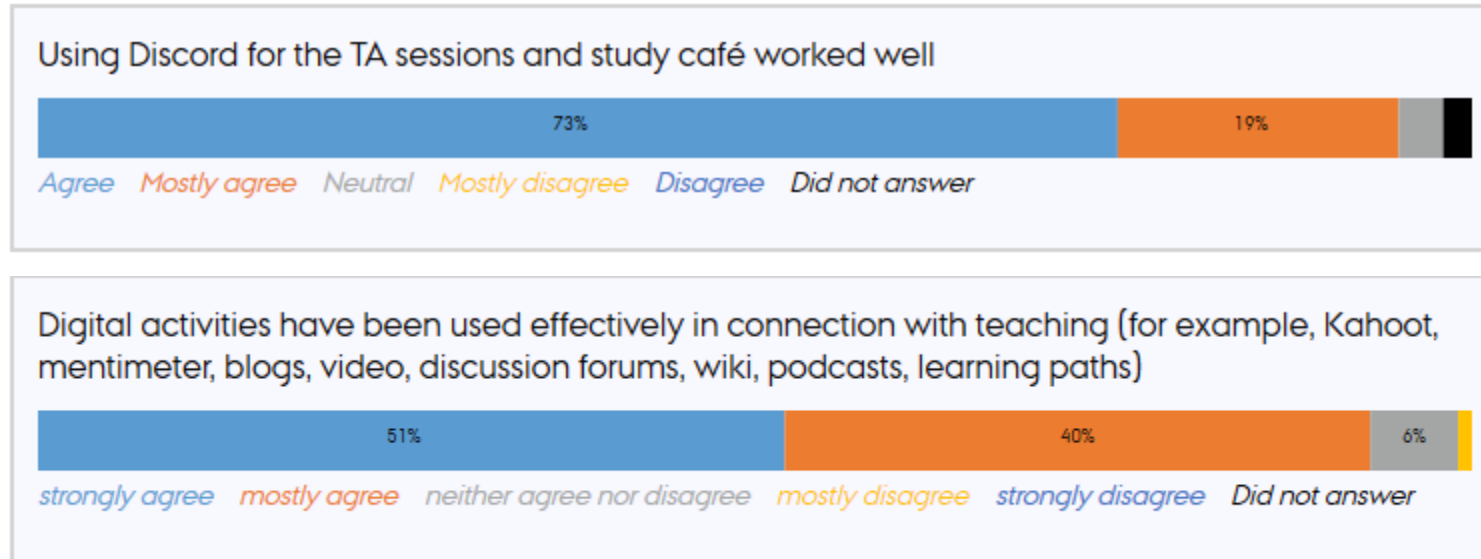
It was important that there were several options for the final project



I had sufficient mathematical background to follow the course



Digital activities



- Would have used the study café more, if it had been on discord from the beginning

Workload

The workload on the course is assessed as



How many hours did you spend on this course altogether (teaching and preparation) per week?



How much time did you on average spend on the handin ?



What fraction of the exercises for the TA sessions did you typically solve ?



- Handin took often a lot of time
- Exercises hard to understand
- Recursion start was tough
- To many evaluation components (handins, project, exam)

TAs - did an excellent job

The student teacher/-s communicated the material in a way that supported my learning - write the student teacher's full name in the comments box



Exam – 26 June 2020

- **6 hours, written exam, with aids, including PC and internet**
- **Communication with others about the exam is not permitted during the exam**
- Reexam in August (format likely the same, but no formal decision)
- 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 will contribute roughly 3/4 to the final grade – but the final grade is an overall assessment
- **eksamen.au.dk**
 - Download .zip + add missing code + upload .zip
- Questions? – post them on Blackboard

Content of .zip file

Exam
questions

test input

test output

```
C.py - C:\Users\au121\Desktop\exam_takehome_example\C.py (3.8.1)
File Edit Format Run Options Window Help

'''
FIBONACCI

Compute the n'th Fibonacci number fib(n), defined recursively:

    fib(0) == 0, fib(1) == 1, fib(n) = fib(n - 1) + fib(n - 2)

Input:

    A single line containing an integer n, 0 <= n <= 10.000

Output:

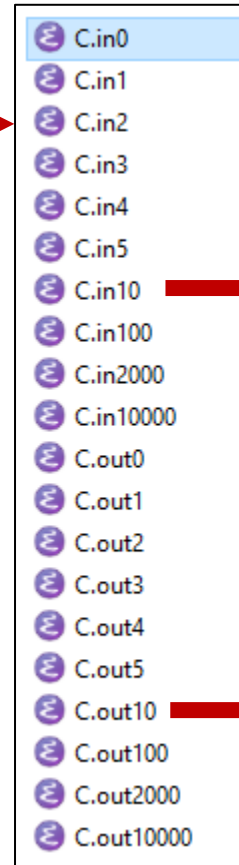
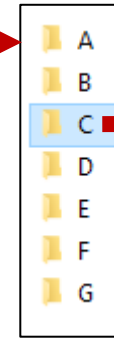
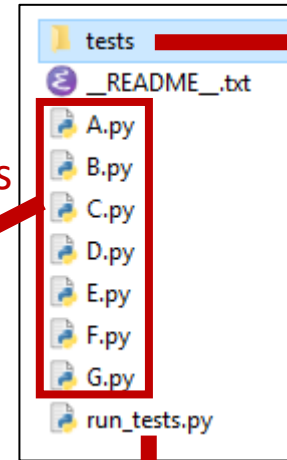
    A single line with the integer fib(n).

Example (see tests/C/... for more examples):

    Input:  10
    Output: 55

'''

n = int(input()) # Read integer n from standard input
answer = None
print(answer)
```



```
C.in10 - Notepad
File Edit Format View Help

10
Ln 1, Col 1
```

```
C.out10 - Notepad
File Edit Format View Help

55
Ln 1, Col 1
```

```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\au121>cd Desktop\exam_takehome_example

C:\Users\au121\Desktop\exam_takehome_example>python run_tests.py C --verbose=0
C.py tests/C\C.in0 [failed]
C.py tests/C\C.in1 [failed]
C.py tests/C\C.in10 [failed]
C.py tests/C\C.in100 [failed]
C.py tests/C\C.in10000 [failed]
C.py tests/C\C.in2 [failed]
C.py tests/C\C.in2000 [failed]
C.py tests/C\C.in3 [failed]
C.py tests/C\C.in4 [failed]
C.py tests/C\C.in5 [failed]

Tests passed:

C      0/10
-----
Total  0/10
=====

C:\Users\au121\Desktop\exam_takehome_example>
```

```
Command Prompt
C:\Users\au121\Desktop\exam_takehome_example>run_tests.py --verbose=-1

Tests passed:

A.py  6/6
B.py  3/6
C.py  0/10
D.py  0/4
E.py  0/6
F.py  0/6
G.py  0/3
-----
Total 9/41
=====
```



```
C.py - C:\Users\au121\Desktop\exam_takehome_example\C.py (3.8.1)
File Edit Format Run Options Window Help

'''
FIBONACCI

Compute the n'th Fibonacci number fib(n), defined recursively:

    fib(0) == 0, fib(1) == 1, fib(n) = fib(n - 1) + fib(n - 2)

Input:

    A single line containing an integer n, 0 <= n <= 10.000

Output:

    A single line with the integer fib(n).

Example (see tests/C/... for more examples):

Input:    10

Output:   55
'''

def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n - 1) + fib(n - 2)

n = int(input()) # Read integer n from standard input

answer = fib(n)

print(answer)
```

Ln: 23 Col: 0

Partial solution that only works for small input

```
Command Prompt
C:\Users\au121\Desktop\exam_takehome_example>run_tests.py C
C.py tests/C/C.in0 [ok]
C.py tests/C/C.in1 [ok]
C.py tests/C/C.in10 [ok]
C.py tests/C/C.in100 [failed]
Input
> 100
Correct output
> 354224848179261915075
Received output
> (none)
Error
> Command '['C:\Users\au121\AppData\Local\Programs\Python\Python38-32\python.exe', 'C.py']' timed out after 5.0 seconds
C.py tests/C/C.in10000 [failed]
Input
> 10000
Correct output
> 336447648764317832666216120051075433103021484606800639065647699746800814421666623681555955136337340255820653326808361593737347
9048386526826304089246305643188735454436955982749160660209988418393386465273130008883026923567361313511757929743785441375213052050
4347701602264758318906527890855154366159582987279682987510631200575428783453215515103870818298969791613127856265033195487140214287
5326981879620469360978799003509623022910263681314931952756302278376284415403605844025721143349611800230912082870460889239623288354
6150577658327125254609359112820392528539343462090424524892940390170623388899108584106518317336043747073790855263176432573399371287
1937587746897479926305837065742830161637408969178426378624212835258112820516370298089332099905707920064367426202389783111470054074
9984592503606335609338838319233867830561364353518921332797329081337326426526339897639227234078829281779535805709936910491754708089
3184105614632233821746563732124822638309210329770164805472624384237486241145309381220656491403275108664339451751216152654536133311
1314042436854805106765843493523836959653428071768775328348234345557366719731392746273629108210679280784718035329131176778924659089
938635459327894523776744061922403376386740040213303432974969020283281459334188268176838930720036347956231171031012919531697946076
3273758925353077255237594378843450406771555577905645044301664011946258097221672975861502696844314695203461493229110597067624326851
5992834709891284706740862008587135016260312071903172086094081298321581077282076353186624611278245537208532365305775956430072517744
3150515396009051686032203491632226408852488524331580515348496224348482993809050704834824493274537326245677558790891871908036620580
0959474315005240253270974699531877072437682590741993963226598414749819360928522394503970716544315642132815768890805878318340491743
4556270520223564846495196112460268313970975069382648706613264507665074611512677522748621598642530711298441182622661057163515069260
0298617049454250474913781151541399415506712562711971332527636319396069028956502882686083622410820505624307017949761711212330660733
10059947366875
Received output
> (none)
Error
> return fib(n - 1) + fib(n - 2)
> [Previous line repeated 995 more times]
> File "C.py", line 24, in fib
> if n <= 1:
> return fib(n - 1) + fib(n - 2)
> RecursionError: maximum recursion depth exceeded in comparison
C.py tests/C/C.in2 [ok]
C.py tests/C/C.in2000 [failed]
Input
> 2000
Correct output
> 422469633339230487870672560234148278257985284025068109801028013731430858437013070722412359963914151108844608753890960360764019
4711643596029271983312598737326253555802606991585915229494253904998722256795316982874482472992263901833716778060607011615497886719
8798583114688708762645973690867228840236544222952433479644801395153495629720876526560695298064998419774487201556128026654045541717
17881930324025204312082516817125
Received output
> (none)
Error
> return fib(n - 1) + fib(n - 2)
> [Previous line repeated 995 more times]
> File "C.py", line 24, in fib
> if n <= 1:
> return fib(n - 1) + fib(n - 2)
> RecursionError: maximum recursion depth exceeded in comparison
C.py tests/C/C.in3 [ok]
C.py tests/C/C.in4 [ok]
C.py tests/C/C.in5 [ok]

Tests passed:

C      7/10
-----
Total 7/10
=====

C:\Users\au121\Desktop\exam_takehome_example>
```

Evaluation of code

- Each problem will be assigned a **weight**
- There will be problems of **varying difficulty**
 - import to be able to differentiate throughout
- 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 fib(n):  
    if n == 10:  
        return 55  
    else:  
        return None
```

AOB ?