

Stærðfræði og reiknifræði – Skilaverkefni 2

Leysið verkefnin með því að færa lausnir hér inn í þessa Júpíter-bók, búa til úr henni PDF-skjal (með *File–Print–PDF* í vafra eða *File–Download As...* í Júpíter) og hlaða því inn í Gradescope. Í Gradescope þarf að tilgreina á hvaða blaðsíðu lausn hvers dæmis er.

Þið megið hjálpast að, en hver fyrir sig verður að skila sinni lausn. Ef þið vinnið náið saman þá þarf að geta vinnufélaga í svari við "Hvernig gekk spurningu aftast" (ekki sleppa henni). Það er bannað að fá lánaðar tilbúnar lausnir eða lána öðrum.

S2.1 Flatarmál og skjölun {-}

Búið til einingu til að reikna flatarmál sbr. tímadæmi T2.3. Hún á að hafa þrjú föll, sem reikna flatarmál hrings ($F = r^2 \pi$), þríhyrnings ($F = gh/2$) og rétthyrnings ($F = ab$). Veljið nöfn á föllin, setjið inn dokkstrengi, vitið eininguna, búið til forrit til að prófa hana og keyrið það. Prófunarforritið á að reikna flatarmál með öllum þremur föllunum og skrifa út ásamt skýringu, t.d. Flatarmál hringsins er 3.45 (með `print(f'Flatarmál hringsins er {fh:.2f}')`). Sýnið líka niðurstöðu úr help-skipun (sbr. T2.3)

In [1]:

```
import math

def circle(r):
    """skilar flatarmáli hrings með  $F=r^2\pi$ """
    F = (r**2)*math.pi
    return F

def triangle(g,h):
    """skilar flatarmáli þríhyrnings með  $gh/2$ """
    F = (g*h)/2
    return F

def square(a,b):
    """skilar flatarmáli rétthyrnings með  $ab$ """
    F = a*b
    return F

print(f'Flatarmal hringsins er {circle(1):.2f}')
print(f'Flatarmal þríhyrningsins er {triangle(5,7):.2f}')
print(f'Flatarmal rétthyrnings er {square(5,5):.2f}')
help(circle)
help(triangle)
help(square)
```

Flatarmal hringsins er 3.14

Flatarmal þríhyrningsins er 17.50

Flatarmal rétthyrnings er 25.00

Help on function circle in module __main__:

```
circle(r)
    skilar flatarmáli hrings með  $F=r^2\pi$ 
```

Help on function triangle in module __main__:

```
triangle(g, h)
    skilar flatarmáli þríhyrnings með  $gh/2$ 
```

Help on function square in module __main__:

```
square(a, b)
    skilar flatarmáli rétthyrnings með  $ab$ 
```

S2.2 Helmingurnarleit {-}

Hér er endurkvæmt reiknirit fyrir helmingurnarleit:

```
a_i...a_(j-1) er raðaður listi
x er stak sem leita skal að
-----
fall leitaij(i,j):
    m = floor((i+j)/2)
    if j <= i, þá skila -1
    annars ef x < a_m þá skila leitaij(i, m)
    annars ef x > a_m þá skila leitaij(m+1, j)
    annars skila m
```

(floor = lækkað niður í næstu heiltölu; notið // í Python). Þetta reiknirit er mjög líkt því sem var á dagskrá í Stærðfræðimynstrum fyrir jóla, með nauðsynlegum breytingum vegna þess að tölusetið er frá 0 til $n - 1$ en ekki frá 1 til n .

Til að hafa einfalda notkun skal hafa fall utanum leitaij með einfaldari fallhaus sem hefur auk þess listann sem leita skal í sem stika:

```
x er stak sem leita skal að
a er listi sem leita skal í, a = [a_0, a_1,...]
-----
fall leita(x,a)
    n = lengd a
    staðs = leitaij(0, n)
    skila staðs
```

Það er sem sé gert ráð fyrir að leitaij sé skilgreint efst í leita, þá getur það haft aðgang að x og a:

```
def leita(x,a)
    """(skjölnun)"""
    def leitaij(...) ...
    ...
    kallað á leitaij(...)
```

1. Skrifðu leita með leitaij í Python. Prófið það fyrir nokkra lista og strengi, gætið þess að öll tilvik í ef-setningunni í leitaij séu reynd.
2. Hægt er ná slembiheiltölu með einingunni random og fallinu randint í henni sbr reit #SLEMBI hér að neðan. Svo er hægt að raða lista með því að nota sort, sbr. grein 1.6.4 í fyrirlestrarnótum. Búið til lista með 1000 slembitölum, raðið honum, og leitið í honum að tölu sem líka er valin af handahófi.

In [1]:

```

import math
import random
import numpy

def leita(x,a):
    """Fallhaus til að einfalda notkun"""
    def leitaij(i,j):
        m = math.floor((i+j)/2)
        #print(m)
        if j <= i: return -1
        elif x < a[m]: return leitaij(i,m)
        elif x > a[m]: return leitaij(m+1,j)
        else: return m
    n = len(a)
    stads = leitaij(0,n)
    return stads

s = "abcdefghijklmnopqrstuvwxyz"
l = range(0,1000000)

# fyrir nedan helmingunarlinu
print(leita(338349,l))
# fyrir ofan helmingunarlinu
print(leita(788987,l))
# strax rett
print(leita(500000,l))
print("Saeti stafs: " + str(leita("t",s)))
# random
x = numpy.random.randint(1000000,size=1000000)
x.sort()
y = random.randint(0,1000000)
print("random: ")
print("random: " + str(leita(y,x)))
help(leita)

```

```

338349
788987
500000
Saeti stafs: 19
random:
random: 744088
Help on function leita in module __main__:

```

```

leita(x, a)
    Fallhaus til að einfalda notkun

```

In []:

```

#SLEMBI
import random
for i in range(0,20):
    x = random.randint(1,5)
    print(x, end=',')

```

S2.3 Listavinnsla {-}

Keyrið reit #EINK til að búa til lista með einkunnum 10 nemenda sem tóku námskeiðið *REI701G Inngangur að listavinnslu*.

1. Skrifðu svo Python-forrit sem skrifar listann út með eftirfarandi sniði

```
Nr Einkunn Ath
1      4.0  F
2      8.0
3     10.0  Ág
...
```

1. Finnið og skrifið út meðaleinkunn og hæstu einkunn og nr. nemanda sem fékk hana (gerið ráð fyrir að það sé bara einn). Ekki gleyma að setja skýringar með útskrifinni. t.d.: `Meðaleinkunn = xx.x` o.s.frv. Skrifðu forritið þannig að það mundi virka þótt listinn `eink` væri styttri eða lengri.
1. Ákvarðið að lokum hve margir fengu hverja einkunn og skrifið út súlurit yfir það með eftirfarandi sniði (bara gefið í heilum):

```
4.0 X
5.0
6.0 XX
7.0 XXXX
8.0
9.0 XX
10.0 X
```

Leiðbeiningar (a) Ekki gleyma að leggja einn við númer nemenda í liðum 1 og 2 því Python byrjar að telja í 0. (b) Notið föllin í grein 1.6.3 í fyrirlestrarnótum til að leysa liði 2 og 3. (c) Til að fá súlu af lengd `k` má nota skipun eins og sýnd er í á borð við `for i in range(k): print('X', end='')` og svo `print()` til að fara í næstu línu, sbr. reit #SÚLA hér fyrir aftan.

In [1]:

```
#EINK
eink = [4.0, 7.0, 10.0, 7.0, 9.0, 7.0, 6.0, 7.0, 9.0, 6.0]
```

In [26]:

```
#SÚLA
k = 5
for i in range(k): print('X', end='')
print()
```

XXXXX

In [2]:

```
#EINK
eink = [4.0, 7.0, 10.0, 7.0, 9.0, 7.0, 6.0, 7.0, 9.0, 6.0]
index = 0
topStudent = max(eink)
indexOfTS = 0
print("Nr.\tEink\tAth")
for x in eink:
    index += 1
    if x >= 9.0:
        print(f'{index}\t{x}\tÁg')
    elif x < 5.0:
        print(f'{index}\t{x}\tF')
    else:
        print(f'{index}\t{x}')
    if x == topStudent:
        indexOfTS = index

avg = sum(eink) / len(eink)
print("Meðaleinkunn: " + str(avg))
print(f'Nemandi nr. {indexOfTS} er með hæstu einkunn: {topStudent}')

fjoldi = "X"
eink.sort()
for x in range(4, 11, 1):
    fj = fjoldi*eink.count(x)
    print(str(x)+".0"+" \t"+str(fj))
```

Nr.	Eink	Ath
1	4.0	F
2	7.0	
3	10.0	Ág
4	7.0	
5	9.0	Ág
6	7.0	
7	6.0	
8	7.0	
9	9.0	Ág
10	6.0	

Meðaleinkunn: 7.2

Nemandi nr. 3 er með hæstu einkunn: 10.0

4.0	X
5.0	
6.0	XX
7.0	XXXX
8.0	
9.0	XX
10.0	X

S2.4 Hvernig gekk?

Skrifið örfá orð aftast í þennan reit um hvernig ykkur gekk að leysa verkefnið. Var það tímafrekt? Of þungt eða of létt? Lærdómsríkt? Með hverjum var unnið? Setjið nafnið ykkar undir.

Þetta gekk bara vel. Ekkert of tímafrekt. Mátulega þungt. Afskaplega skemmtilegt. Python er snilld. Ég vann þetta bara sjálfur. -Teitur Guðmundarson