MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Kursledare: Anders Mörtberg
Examinator: Lars Arvestad

Tentamensskrivning i
DA2004 Programmeringsteknik för matematiker
7.5 hp
2021-04-21

- Del A har flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt rätt antal alternativ får man 0 poäng på frågan.
- Man måste bli godkänd på del A (4 rätt på 8 frågor) för att få göra del B.
- Del B består av ett antal frågor med varierande poäng (totalt 12) vilka ska lösas genom att man skriver Python 3 kod.
- Svaren till del B lämnas in i en `.py`-fil namngiven `anonymkod.py` där `anonymkod` är koden som man får i Ladok när man registrerar sig på tenta. Man **måste** även ha med sin anonyma kod i en kommentar i toppen av filen. Man ska **inte** skriva sitt riktiga namn någonstans i filen!
- Var *noga* med att **namnge** *funktioner och klasser* rätt (på samma sätt som de är namngivna i uppgiften).
- Inga `import` får användas om de inte nämns eller finns med i uppgiften. Man får dock använda inbyggda funktioner som `len`, `range` och `map`.
- All kod avser **Python 3**, dvs *inte* t.ex. Python 2.7
- **Hjälpmedel:** Till del A får man ha ett A4 med så mycket information man vill. Du får skriva på båda sidorna. Del B är öppen bok då det är hemtenta och samma regler kring hjälpmedel gäller som för projekt och labbar.
- **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.

Del A: flervalsfrågor (1p per fråga)

1. Vilka av följande bör `assert` användas för?

A. Kontrollflöde
B. Användarinteraktion
C. Felsökning
D. Göra Pythonkod snabbare
E. Ingenting, det finns inget som heter `assert` i Python

2. Vilka av följande ord har specifikt med filhantering att göra?

A. `open`
B. `for`
C. `exit`
D. `close`
E. `continue`

3. Vad blir resultatet av `[ [ x[i] for x in [[1, 2, 3], [4, 5, 6]] ] for i in range(3)]`?

A. `[[0, 1, 2], [0, 1, 2]]`
B. `[]`
C. `[[1, 2, 3], [4, 5, 6]]`
D. `[[1, 4], [2, 5], [3, 6]]`
E. `[1, 2, 3, 4, 5, 6]`

4. Vilka värden skrivs ut av koden till höger?

A. `'a'` och `'b'`

B. `[0, 1, 2]` och `[3, 4, 5]`

C. `'a'`, `[0, 1, 2]`, `'b'` och `[3, 4, 5]`

D. `{ 'a': [0, 1, 2], 'b': [3, 4, 5] }`

E. Inga, man får inte ha listor i uppslagstabeller

```
d = { 'a': [0, 1, 2], 'b': [3, 4, 5] }

for x in d:
    print(d[x])
```

5. Betrakta koden till höger, vad returneras om man kör `f()`?

A. `10`

B. `11`

C. `9`

D. `None`

E. `8`

```
a = 2
b = 3
c = 4

def f(a = 1):
    b = 5
    return (a + b + c)
```

6. Givet koden nedan, vad blir värdet på `result`?

A. `None`

B. `'[]""'`

C. `'howareyou?'`

D. `'how are you ?'`

E. `'howare?'`

```
mylist = [['how'], '[are]', 'you', '"?"']
result = ''

for x in mylist:
    out = ''
    for c in x:
        if c not in '[]"':
            out += c
    result += out
```

7. Vad returneras av anropet `fcn(0)` givet definitionen nedan?

```
import random

def fcn(param):
    x = 0
    while random.random() > 0.5:
        if param > x:
            x += 1
            continue
        else:
            break
    return x
```

A. `0`

B. `1`

C. `2`

D. `10`

E. Det går inte att förutsäga då det är slumpmässigt

8. Givet funktionen nedan, vad blir resultatet av `d([0, [1], [2, 3], [[4, 5]]])`?

A. `0`

B. `1`

C. `2`

D. `3`

E. `None`

```
def d(l):
    if type(l) == list:
        return 1 + max(map(d, l))
    else:
        return 0
```

Del B: kodfrågor (2p per fråga)

Obs: ni får använda `random` biblioteket på den här delen.

9. På den ordinarie tentan 2021-03-12 handlade en uppgift om att implementera en klass `Dice` för sexsidiga tärningar. En möjlig lösning följer nedan:

```
class Dice:

    def roll_die(self):
        return random.randint(1,6)

    def roll_dice(self,n):
        sum = 0
        for i in range(n):
            sum += self.roll_die()
        return sum
```

Utöka koden för `Dice` med en konstruktor som låter oss sätta `number_of_sides` som säger hur många sidor tärningen har så att vi kan få tärningar med andra antal sidor än `6`. De möjliga antalet sidor vi ska kunna ha är `4`, `6`, `8`, `12`, och `20`. Om användaren inte anger en specifik storlek på tärningen ska en sexsidig tärning skapas. Om användaren ger något annat än de godkända tärningstyperna ska ett lämpligt särfall lyftas.

Metoderna `roll_die` och `roll_dice` ska även modifieras så att de fungerar för de nya tärningsstorlekarna.

Exempel: koden nedan:

```
d = Dice()
print(d.roll_die())
print(d.roll_dice(2))

t20 = Dice(20)
print(t20.roll_die())
print(t20.roll_dice(2))
```

ger följande möjlig utdata:

```
4
7
16
27
```

Obs: då tärningskast är slumpmässiga kommer ni troligtvis få andra utdata än de ovan.

10. Inspirerad av koden för `Dice` skriv en klass `ArbDice` (för ”arbitrary dice”) där användaren även kan ange vilka värden sidorna ska ha. Detta görs genom att användaren även ger en lista med lika många värden som tärningen har sidor. Dessa ska sedan sparas i en uppslagstabell från sidnummer (tal mellan `1` och `number_of_sides`) till värdet på den sidan. Utelämnas listan med nya värden ska värdena `1` till `number_of_sides` sättas i uppslagstabellen.

Metoderna `roll_die` och `roll_dice` ska även modifieras så att de fungerar för tärningar med godtyckliga sidor. Ett lämpligt särfall ska även lyftas om användaren ger en lista med fel antal sidovärden.

Exempel: koden nedan:

```
d4 = ArbDice(4,[10,20,30,40])
print(d4.roll_die())
print(d4.roll_dice(2))

d6 = ArbDice()
print(d6.roll_die())
print(d6.roll_dice(2))
```

ger följande möjlig utdata:

```
30
80
1
9
```

11. Skriv en funktion `mask(s1,s2)` som tar in två strängar `s1` och `s2` som byter ut bokstäverna till `*` i `s1` i den ordning de förekommer i `s2`, returnerat i en ny sträng.

    Obs: gemener och versaler ska inte spela någon roll

    Exempel:

    ```
    [In] : print(mask('Hello, how are you Anders?', 'ehoaar'))
    [Out]: H*llo, **w *re you *nde*s?
    [In] : print(mask('Hello, how are you Anders?', 'H,HAAR?A'))
    [Out]: *ello* *ow *re you *nde*s*
    ```

    Notera att i det första exemplet byts inte `o` ut i `you` då det redan använts för att byta ut `o` i `how`.

12. Ett sätt att kryptera en text är att skriva in alla bokstäver i en matris med bestämt antal kolumner. Alla bokstäver fylls i rad för rad och sen läser man ut det krypterade ordet genom att läsa bokstäverna kolumnvis.

    Exempel: för att kryptera `'Secret text'` med `3` kolumner skapar vi matrisen:

`'S'`	`'e'`	`'c'`
`'r'`	`'e'`	`'t'`
`' '`	`'t'`	`'e'`
`'x'`	`'t'`	

    Den krypterade texten är då `'Sr xeettcte'`.

    Skriv en funktion `matrix_encrypt(s,cols)` där `s` är en sträng och `cols` är ett positivt heltal som representerar hur många kolumner matrisen ska ha, och den krypterade texten returneras.

    Tester:

    ```
    [In] : print(matrix_encrypt('Secret text',3))
    [Out]: Sr xeettcte
    [In] : print(matrix_encrypt('Secret text',2))
    [Out]: Sce eterttx
    [In] : print(matrix_encrypt('Secrettext',5))
    [Out]: Stetcerxet
    ```

    Obs: om längden på `s` inte är jämnt delbar med `cols` så ska man inte få några extra mellanslag eller andra tecken i slutet av den krypterade strängen.

13. Skriv en funktion `matrix_decrypt(s,cols)` för matrisdekryptering. Det vill säga, given krypterad sträng `s` och ett heltal `cols`, returnera ordet i klartext.

    Tester:

    ```
    [In] : print(matrix_decrypt('Sr xeettcte',3))
    [Out]: Secret text
    [In] : print(matrix_decrypt('Sce eterttx',2))
    [Out]: Secret text
    [In] : print(matrix_decrypt('Stetcerxet',5))
    [Out]: Secrettext
    ```

14. Skriv en högre ordningens funktion `encrypt_file(f,enc)` som tar in en sträng `f` och en krypteringsfunktion `enc`. Varje rad i filen `f.txt` ska sedan krypteras med hjälp av `enc` och resultatet ska skrivas till `f_encrypted.txt`.

    Exempel: om filen `myfile.txt` innehåller

    ```
    secret texts
    writteninmy
    secret files
    ```

    och man kör `encrypt_file('myfile',lambda x: matrix_encrypt(x,3))` så ska filen `myfile_encrypted.txt` innehålla:

    ```
    sr xeettctes
    wtnmrtiyien
    sr leefectis
    ```