

Tentamen: Programmering med PYTHON, MVG301.

Tid och plats: Fre 18/8 2023 kl 08:30-12:30

Ansvarig lärare: Katarina Blom. Anknytning 1097

Betygsgränser: G: 12p, VG: 18p, maximalt 24p

Hjälpmedel: Inga hjälpmedel. Alla lösningar ska skrivas i PYTHON, kommentera gärna koden kortfattat.

- 1 Två primtal p och q är primtalstvillingar om $q - p = 2$. Skriv ett program i PYTHON (4p) som skriver ut alla primtalstvillingar mindre än 100. Använd gärna funktionen `isprime` från paketet `sympy` i din lösning. Anropet `isprime(n)` returnerar `True` om talet n är ett primtal, `False` om talet n inte är ett primtal.

- 2 Om man har en text kan man kryptera den genom att ersätta bokstäverna i texten med (4p) bokstäver ett visst antal steg längre fram i alfabetet (det engelska alfabetet 'A':'Z'). Om vi har texten 'VIRYMMERINATT' och ersätter bokstäverna med bokstäver 1 steg längre fram i alfabetet får vi 'WJSZNNFSJOBUE'.

Skriv en funktion i PYTHON

`kryptera(wrd, n)`

som har två inparametrar, `wrd` ett ord (en sekvens av bokstäver) och ett tal n . Funktionen ska returnera strängen där alla bokstäver i ordet bytts ut mot bokstäver n steg längre fram i alfabetet. Om n är för stort, så att bokstäverna hamnar 'utanför' alfabetet börjar man räkna från början av alfabetet igen, tex anropet

`kryptera('VIRYMMERINATT', 27)` returnerar 'WJSZNNFSJOBUE'

och anropet

`kryptera('VIRYMMERINATT', 5)` returnerar 'ANWDRRJWNSFYU'

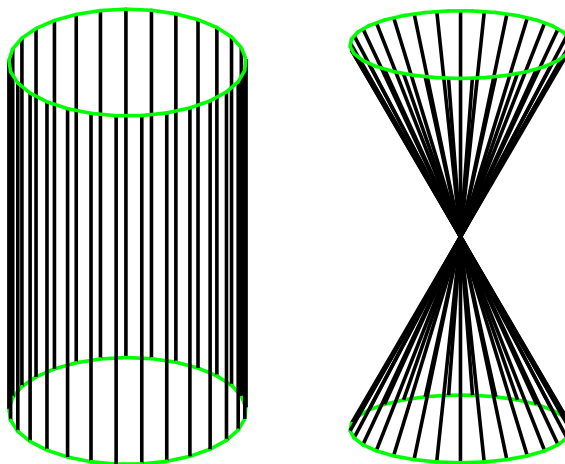
Du kan anta att ord består av enbart versaler (stora bokstäver) i det engelska alfabetet ('A':'Z'). Om $n \leq 0$ returneras ordet okrypterat.

- 3 En Toplitzmatris är en så kallad diagonalkonstant matris, dvs alla elementen i en diagonal (4p) (från vänster till höger) är lika. Följande matriser är Toplitzmatriser:

$$\begin{bmatrix} a & b & c \\ f & a & b \\ e & f & a \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 & 7 \\ 3 & 1 & 0 & 2 \\ 7 & 3 & 1 & 0 \\ 9 & 7 & 3 & 1 \end{bmatrix}, \begin{bmatrix} a & b & c \\ f & a & b \end{bmatrix}$$

Skriv en funktion `toplitz(M)` som tar en matris M som argument och som returnerar `True` om M är en Toplitzmatris.

- 4 I figuren nedan till vänster ser vi en cylinder. För att rita figuren har man ritat en cirkel (4p) i nivån $z = 0$ och en cirkel i nivån $z = 1$ (de två gröna cirkelarna). Sedan har man ritat 30 räta linjer mellan de två cirkelarna. I den högra figuren har man vridit taket på cylindern π radianer. Skriv en sekvens i PYTHON som ritat figurerna. (Använd `plot3D` i paketet `matplotlib.pyplot` för att rita cirkelarna och linjerna). Observera att färgerna i figuren är inte viktiga.



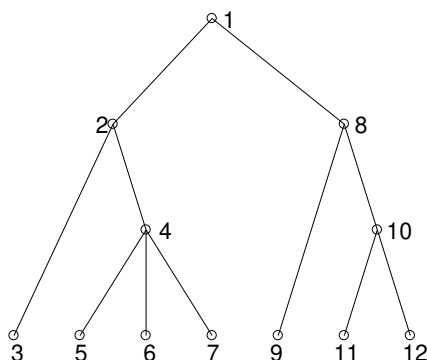
- 5 I textfilen `dig.txt` ska det finnas exakt ett kommatecken (',') per rad. Men det finns (4p)
rader i filen där detta inte stämmer. Skriv en funktion i PYTHON som beräknar (och
returnerar) vilka rader i filen `dig.txt` som inte innehåller exakt ett kommatecken. Anta
att filen `dig.txt` finns i aktuell katalog.
- 6 Trädet nedan innehåller noderna 1, 2, 8, 4, 10 och löven 3, 5, 6, 7, 9, 11, 12. För att (4p)
representera trädet i PYTHON har man använt en vektor `nodes` som innehåller varje nods
(och lövs) förälder. Plats nummer j ($j > 0$) i `nodes` innehåller j :s förälder i trädet enligt
följande:

```
nodes[0] = 12 # Elementet 0 i vektorn anger antal noder och löv i trädet
nodes[1] = 0  # Element 1 i vektorn (roten) har 0 som förälder

nodes[2] = 1  # element nummer 2 har 1 som förälder (roten)
nodes[3] = 2  # element nummer 3 har 2 som förälder
...
nodes[12] = 10 # element nummer 12 har 10 som förälder
```

Trädet i figuren representeras av vektorn

```
nodes = [12, 0, 1, 2, 2, 4, 4, 4, 1, 8, 8, 10, 10]
```



Skriv en funktion i PYTHON som har två parametrar, ett träd (lagrat i en vektor enligt beskrivning ovan) och ett tal. Om talet är ett löv i trädet ska funktionen returnera vägen från roten till lövet. Om talet inte är ett löv i trädet returneras en tom väg. (T.ex. vägen från roten till lövet 7 är $[1,2,4]$).

Du kan utgå från att funktionen anropas med ett giltigt träd, och att andra parametern (talet) alltid är ett heltal.

Lycka till!!

Inbyggda typer och operatörer	
Heltal, decimaltal: int, float round(x), round(x,n) abs(x)	Operatörer *, /, +, -, **, %, // och >, <, >=, <=, ==, != Avrundar till närmsta heltal respektive till n decimaler x
Komplexa tal: complex z = a+bj z.real, z.imag	Operatörer *, /, +, -, abs skapar $z = a + bj$, a, b heltal eller reella tal realdel respektive imaginärdel av z
Sanningsvärden: bool, True/False	Operatörer and, or, not samt de för int ovan
Tecken: str s = "ABC" ord(c), chr(n) s.upper(), s.lower() s1 + s2 s.replace("från","till") s.splitlines(), s.split() L.join(t)	Operatörer >, <, >=, <=, ==, !=, + skapar textsträngen (str) med bokstäverna ABC teckenkod för tecknet c, tecknet med teckenkoden n stora/små bokstäver i s konkatenerar textsträngarna s1 och s2 ersätter förekomster av från med till i s delar upp s i delsträngar konkatenerar strängarna i sekvensen L med tecknet t mellan
Sekvenser: list, tuple, range, str L = [1,2,3] T = (1,2,3) r = range(3) r = range(st,sl,steg) x in L, x not in L [x] + L L*n L[i] L[i:j] L[i:j:k] L[::-1] len(L) zip(L1,L2) max(L), min(L) L.index(x) L.count(x)	Nedan är L sekvens och x ett element: skapar listan [1, 2, 3] skapar tupeln (1, 2, 3) skapar sekvensen 0, 1, 2 element från och med st, till sl med steget steg ger True respektive False om x finns i L konkatenerar (sätter ihop) listorna [x] och L konkatenerar L n gånger i:e elementet i L elementen från och med i till j i L elementen från och med i till j med steget k elementen i L sorterade baklänges antal element i L kombinerar elementen i sekvenserna L1 och L2 största/minsta element i L index för första förekomsten av x i L antal förekomster av x i L.
Avbildningar: dict D = {'A':1, 'B':2, 'C':3} D[key] L = D.keys() L = D.values() L = D.items() in, not in, max, min, zip, len	lista med (nyckel,värde)-par avbildning med nycklar A, B, C och värden 1, 2, 3 värdet med nyckeln key ger en lista med nycklarna en lista med värdena ger en lista med (nyckel,värde) par se sekvenser ovan
Typer övrigt: Typomvandla med typnamn(variabel), t.ex int(x), list(s), dict(zip(L1,L2)) type(x) isinstance(x, typnamn)	ger typen på x True om x tillhör typen typnamn

Kontrollstrukturer och egendefinerade funktioner			
if-sats	while-loop	for-loop	funktioner
if villkor: satser elif villkor: satser else: satser Man kan ha flera elif elif och else kan utelämnas villkor är ett logiskt villkor	while villkor: satser villkor är ett logiskt villkor break bryter en loop	for e in L: satser L är en sekvens, dict eller np.array	def namn(p): satser return värde värde är funktionens returvärde, p är inparametrar Man kan ha flera return return kan utelämnas

Klasser	
class namn: def __init__(self,p): def __str__(self): def __add__(s1,s2):, def __sub__(s1,s2): def __mul__(s1,s2):, __truediv__(s1,s2):	skapa klass med namnet namn (om)definera konstruktor med parameterlista p (om)definera utskriftsfunktion överlagra +, - överlagra *, /

I/O (nedan är s en str, l en lista av str och v1, v2 godtycklig typ)	
<pre>print(v1), print(v1,v2) s = input('text') fid = open('fil','spec') fid.close() with open('fil','spec') as fid: s = fid.read() l = fid.readlines(), s = fid.readline() fid.write(s)</pre>	<p>skriver innehållet i v1, v2 på terminalen</p> <p>läser in från tangentbordet</p> <p>öppna fil, spec är r eller w</p> <p>stänger filen som fid refererar till</p> <p>öppna fil, spec är r eller w</p> <p>läser innehållet i filen</p> <p>läser alla rader, en rad i filen</p> <p>skriver s till filen</p>
<pre>f'{v1:format} {v2:format}' format kan vara nd (int), n.de (float med d decimaler på exponentiell form) n.df(float med d decimaler). n är antalet positioner talet ska ta i anspråk. '\n' nyradstecken</pre>	<p>Bilda en formaterad sträng av v1, v2</p>

Operativsystemskommandon	import os
l = os.listdir(path)	ger en lista av filer och kataloger i path
os.chdir(path)	byter aktuell katalog till path
os.getcwd()	visar aktuell katalog
l = os.walk(path)	en lista av sökvägar, kataloger och filer under path

Några funktioner/konstanter i math	
x är en skalär av typen float eller int	from math import *
sin(x), cos(x), tan(x), atan(x), cosh(x), sinh(x)	sin(x), cos(x), tan(x), arctan(x), cosh(x), sinh(x)
exp(x), log2(x), log10(x), log(x)	e^x , $\log_2(x)$, $\log_{10}(x)$, $\ln(x)$
sqrt(x), pi, e	\sqrt{x} , π , e

Några funktioner/konstanter i NumPy	
L är en lista (list), A, B, C är np.array	import numpy as np
<pre>A = np.array(L) A = np.linspace(st,sl,antal) A = np.arange(st,sl,steg) A = np.reshape(L, (r, k)) C = np.append(A, B) A = np.zeros([r,k]), A = np.ones([r,k]) A = np.random.random([r,k]) [r,k] = np.shape(A), n = np.size(A) e = A[r,k], A[i,:], A[:,j] A==B, A==e, A>B, A>e ... A+B, A+e, A*B, A*e ... np.all(A==e), np.any(A==e) np.sin(A), np.cos(A), np.tan(A), np.atan(A) np.cosh(A), np.sinh(A) np.exp(A), np.log2(A), np.log10(A), np.log(A) np.sqrt(A), np.pi, np.e np.round(A) C = np.matmul(A,B), C = np.dot(A,B) C = np.transpose(A) np.max(A), np.min(A), np.sum(A) A = np.load('filnamn'), np.save('filnamn',A)</pre>	<p>np.array med elementen i L</p> <p>antal element på intervallet [st,sl]</p> <p>element från och med st till sl med steget steg</p> <p>r×k matris med elementen i L</p> <p>konkatenera A och B</p> <p>r×k nollor, ettor</p> <p>r×k slumpstal</p> <p>antal rader och kolumner i A, antal element i A</p> <p>Element på positionen r,k, rad i, kolumn j, i A</p> <p>Jämförelseoperatorerna opererar elementvis</p> <p>Aritmetiska operatorer opererar elementvis</p> <p>True om alla elem i A är e, något elem i A är e</p> <p>sin(A), cos(A), tan(A), arctan(A) (elementvis)</p> <p>cosh(A), sinh(A) (elementvis)</p> <p>e^A, $\log_2(A)$, $\log_{10}(A)$, $\ln(A)$ (elementvis)</p> <p>\sqrt{A} (elementvis), π, e</p> <p>avrunda till närmsta heltal (elementvis)</p> <p>matrismultiplikation, skalärprodukt</p> <p>transponat $C=A^T$</p> <p>största/minsta element, summan av elementen i A</p> <p>laddar in data, sparar data på fil</p>

Några funktioner i Matplotlib.pyplot	
x, y, z är list eller np.array	import matplotlib.pyplot as plt
<pre>plt.plot(x,y) plt.plot(x,y,'egenskaper') plt.fill(x,y,'f') plt.xlabel('text'), plt.ylabel('text') plt.title('text'), np.grid() plt.axis('equal'), plt.axis('off') ax = plt.axes(projection='3d') ax.plot3D(x,y,z) X,Y=np.meshgrid(x,y) ax.plot_surface(X,Y,Z)</pre>	<p>sammanbinder punkterna (x_i, y_i) med rätta linjer</p> <p>egenskaper är t.ex. färg, typ av linje.</p> <p>yller området som innesluts av (x_i, y_i) med färgen f</p> <p>titel på x-, y- axel</p> <p>figurtitel, gridlinjer</p> <p>skala lika långt i x- och y-led, ta bort axlarna.</p> <p>skapar koordinatsystem i rummet</p> <p>sammanbinder punkterna (x_i, y_i, z_i) med rätta linjer</p> <p>ordna gridpunkter</p> <p>skapar funktionsytan $\{(x, y, z) : z = f(x, y)\}$</p>
Vanliga färger: r (röd), b (blå), g (grön), k (svart), w (vit), y (gul), m (magenta), c (cyan)	
Vanliga linjeegenskaper: -, :, .-, - -, vanliga punktmarkörer, *, o, +, x	