Oppgave 1 - Flervalgsoppgaver

Hva er 5310 i det hexadesimale tallsystemet?

**Svar:** 3516

Hvilken av følgende påstander er korrekte?

**Svar:** En IP-adresse identifiserer ikke en spesifikk datamaskin, men en forbindelse mellom en datamaskin og et nettverk. En ruter har en IP-adresse for hver nettverks-forbindelse.

Hva er sant om PandA?

**Svar:** PandA sier om et fenomen er tilstede eller ikke på et bestemt sted og tid

Hva er de tre vanligste overføringsfeilene mtp. dataoverføringer?

**Svar:** Interference (forstyrrelse), distortion (forvrenging), attenuation (svekking)

Ved linjesvitsjing...

**Svar:** ...opprettes et kommunikasjonssignal mellom en sender og en mottaker når det er nødvendig, og forsvinner etter bruk. Denne veien kommer ikke i kontakt med andre signaler (isolert vei).

Hva gjør nettverkslaget (IP-layer)?

**Svar:** Spesifiserer formatet på pakkene som sendes over nettet, og mekanismene for å sende pakker fra én datamaskin via rutere, til en annen.

Oppgave 2 - Kodeforståelse

**a)**answer = 42

Funksjonen itererer gjennom en todimensjonal liste og vil for hver liste i den todimensjonale listen legge strengen av ett element i listen til s, hvor hvilket elementet som legges til s bestemmes av lengden til listen.

**b)** Utskrift til skjerm:

1 100  
4 10  
5 1

Funksjonen finner hvor mange 1'ere, 10'ere, 100'ere, osv. det er i tallet (argumentet) og printer dette ut til skjerm med den største verdien først.

**c)**answer = 'PandA'

Funksjonen tar inn en liste med strenger og en karakter, itererer over listen og returnerer en streng bestående av den første bokstaven i strengene med karakteren i (enten stor eller lav).

**d)**Kodesnutt 5.

Oppgave 3 - Tautonymer

**Oppgave 3a**

|  |
| --- |
| def isTautonym(s):      lst = s.split()      if len(lst) < 2:          return False      el = lst[0].lower()      for i in range(1,len(lst)):          if el != lst[i].lower():              return False      return True |

Oppgave 4 - Lister

**Oppgave 4a**

|  |
| --- |
| #Det er flere måter å løse denne oppgaven på  #Alternativ 1 - while-løkke  def num\_el(el,lst):      lst2 = lst.copy()      r = 0      while el in lst2:          r += 1          lst2.remove(el)      return r    #Alternativ 2 - innebygde funksjoner  def num\_el(el,lst):      return lst.count(el)    #Alternativ 3 - for-løkke  def num\_el(el,lst):      r = 0      for ele in lst:          if ele == el:              r += 1      return r |

**Oppgave 4b**

|  |
| --- |
| #Det er flere måter å løse oppgaven på  #Alternativ 1 - for- og while-løkke  def remove\_duplicates(lst):      for el in lst:          while num\_el(el,lst)>1:              lst.remove(el)      return lst    #Alternativ 2 - dobbel for-løkke  def remove\_duplicates(lst):      for el in lst:          for i in range(1,num\_el(el,lst)):              lst.remove(el)      return lst |

**Oppgave 4c**

|  |
| --- |
| #Under er to mulige måter å løse oppgaven på  #Alternativ 1  def sorted\_list(s):      lst = s.split()      lst2 = []      while len(lst):          el = lst[0]          for ele in lst:              if ele.lower()<el.lower():                  el = ele          lst.remove(el)          lst2.append(el)      return lst2    #Alternativ 2  def sorted\_list2(s):      liste = s.split()      unsorted = 1             # Husk at 1 = True      while unsorted:          unsorted = 0         # 0 = False          for x in range(0, len(liste)-1):              if (liste[x].lower() > liste[x+1].lower()):                  liste[x], liste[x+1] = liste[x+1], liste[x]                  unsorted = 1      return liste |

Oppgave 5 - Quiz

**Oppgave 5a**

|  |
| --- |
| def begins\_with(capitals, letter):      result = []      for city in capitals:          if city[0] == letter:              result.append(city)      return result |

**Oppgave 5b**

|  |
| --- |
| def random\_latin\_letter(upper):      import random      if upper:          ascii\_value = random.randint(65,90)      else:          ascii\_value = random.randint(97,122)      letter = chr(ascii\_value)      return letter |

**Oppgave 5c**

|  |
| --- |
| def letter\_question(capitals):      n\_solutions = 0      while n\_solutions == 0:          letter = random\_latin\_letter(True)          solution = begins\_with(capitals, letter)          n\_solutions = len(solution)      question = 'Capital starting with ' + letter + '? '      return question, solution |

**Oppgave 5d**

|  |
| --- |
| def n\_letters(city):      length = len(city)      for i in city:          if city[i] == " ":              length -= 1      return length    Ans:  Nei, gir IndexError: string index out of range |

**Oppgave 5e**

|  |
| --- |
| def has\_length(capitals, length):      result = []      for city in capitals:          if n\_letters(city) == length:              result.append(city)      return result |

**Oppgave 5f**

|  |
| --- |
| def length\_question(capitals):      import random      ok\_lengths = []      for city in capitals:          length = n\_letters(city)          if length not in ok\_lengths:              ok\_lengths.append(length)      num = random.choice(ok\_lengths)      question = 'Capital with ' + str(num) + ' letters? '      solution = has\_length(capitals, num)      return question, solution |

**Oppgave 5g**

|  |
| --- |
| def random\_question(capitals):      import random      q\_type = random.randint(0,1)      if q\_type == 0:          que, sol = letter\_question(capitals)      else:          que, sol = length\_question(capitals)      return que, sol |

**Oppgave 5h**

|  |
| --- |
| def play\_quiz(capitals):      keep\_on = True      points = 0      while keep\_on:          quest, sol = random\_question(capitals)          city = input(quest)          if city == 'Q':              keep\_on = False          elif city in sol:              print('Correct!')              points += 1          else:              print('Sorry, that was wrong!')      return points |

**Oppgave i**

|  |
| --- |
| def main():      print('CAPITALS QUIZ')      print('Answer questions to score points.')      print('To quit, answer Q')      points = play\_quiz(CAPITALS)      if points == 1:          p\_text = '1 point.'      else:          p\_text = str(points) + ' points.'      print('Congrats! You scored', p\_text) |

**Oppgave j**

|  |
| --- |
| def make\_question(row,col,inverse):      type = QUIZ\_INFO[0][col]      country = QUIZ\_INFO[row][0]      what = QUIZ\_INFO[row][col]        if inverse:            fasit = country          question = "In which country is " + what + " the " + type      else:          fasit = what          question = "What is the " + type + " in " + country        return question, fasit |

BONUSOPPGAVE:

**Bonus**

|  |
| --- |
| def pythagoranTriples(n):      triplets = []      a = 2      count = 0      while count < n:          maxsearch = a\*\*2 #vilkårlig høyt tall slik at b^2-(b-1)^2 > a          for b in range(a,maxsearch):              c = (a\*\*2 + b\*\*2)\*\*(1/2)              if c%1 == 0:                  triplets.append([a, b, int(c)])                  count +=1          a += 1      return triplets |