1 Problem with solution 47

1.1 TSNE representation

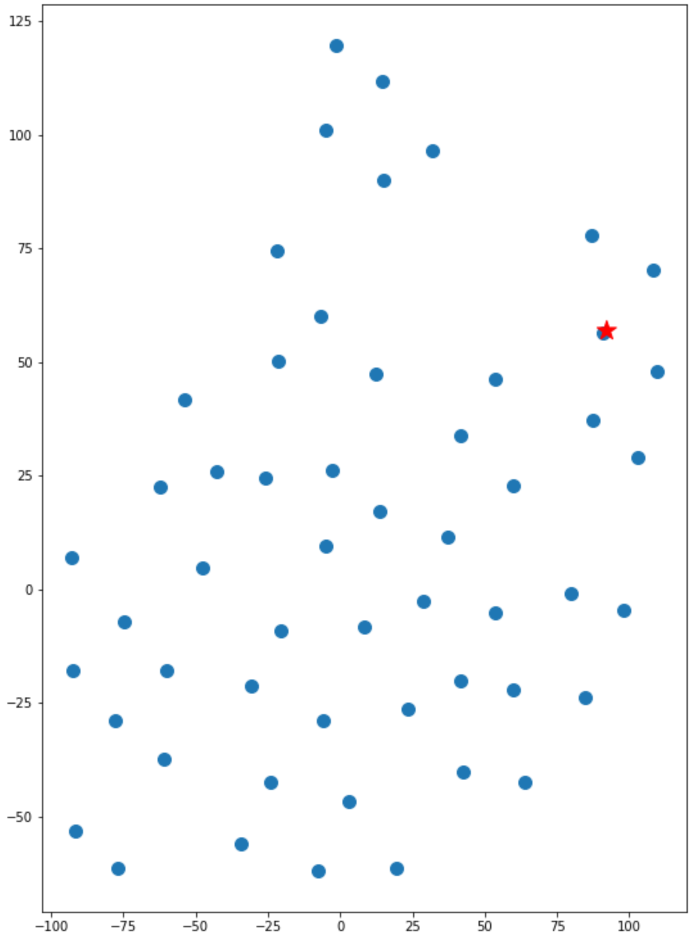

Figure 1: TSNE 2d feature space

1.2 Problem and Solution

Cluster: 2
Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.
Suppose the following input is supplied to the program:
9
Then, the output should be:
11106

```
# Complete your function using this header
def digit_sum(digit):
    total = 0
    for i in range(1,5):
        number = "%s" % digit
        number = int(number * i)
        total = total + number
    return total
```

2 Problem with solution 11

2.1 TSNE representation

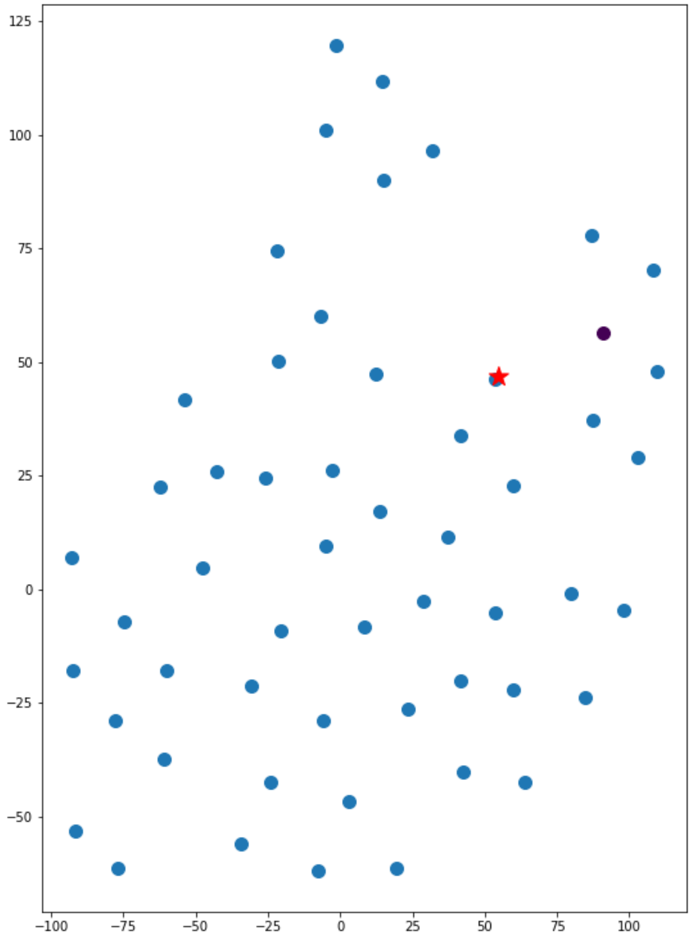

Figure 2: TSNE 2d feature space

2.2 Problem and Solution

Cluster: 2
Write a function that takes as argument a word and returns (boolean type True or False) whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

```
def reverse(word):
    x = ''
    for i in range(len(word)):
        x += word[len(word)-1-i]
    return x == word
```

3 Problem with solution 17

3.1 TSNE representation

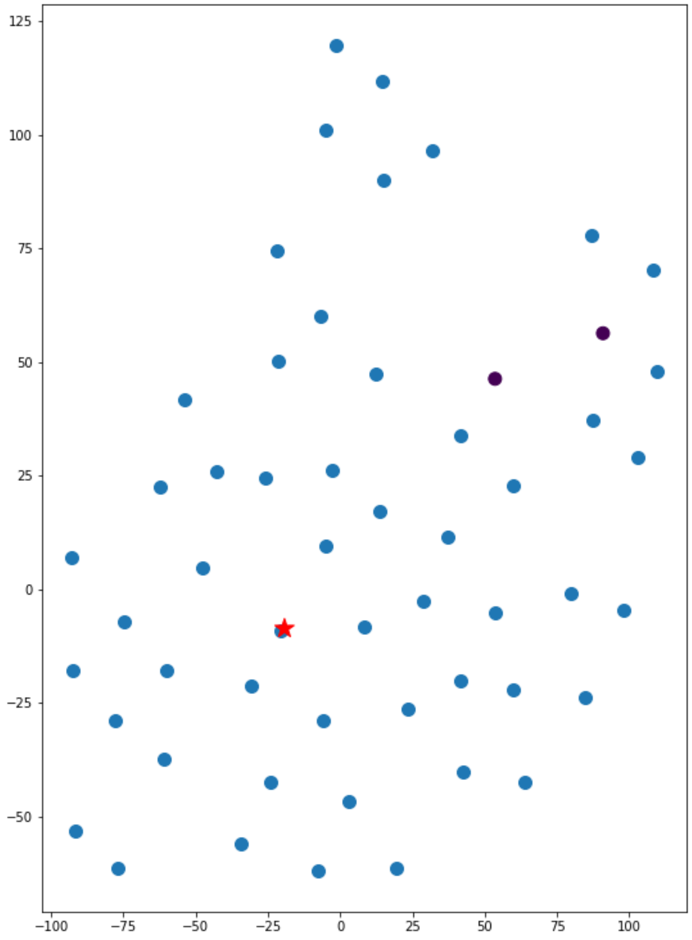

Figure 3: TSNE 2d feature space

3.2 Problem and Solution

Cluster: 2
Write a function that receives as an argument how old the user is. Return a sentence saying "You are old enough to vote." if they are over 16 years old or "You can vote in X years.", substituting the X with the appropriate number of years. Add another sentence if the user is younger than 65 years old, calculating the number of years missing until they can retire (assume at age 65): "You can retire in Y years.".

```
def vote_and_retire(age):
    sentence = ""
    if age >= 16:
        sentence = "You are old enough to vote."
    else:
        ageToVote = 16 - age
        sentence = "You can vote in {0} years.".format(ageToVote)

    if age < 65:
        ageToRetire = 65 - age
        sentence += " You can retire in {0} years.".format(ageToRetire)

    return sentence
```

4 Problem with solution 34

4.1 TSNE representation

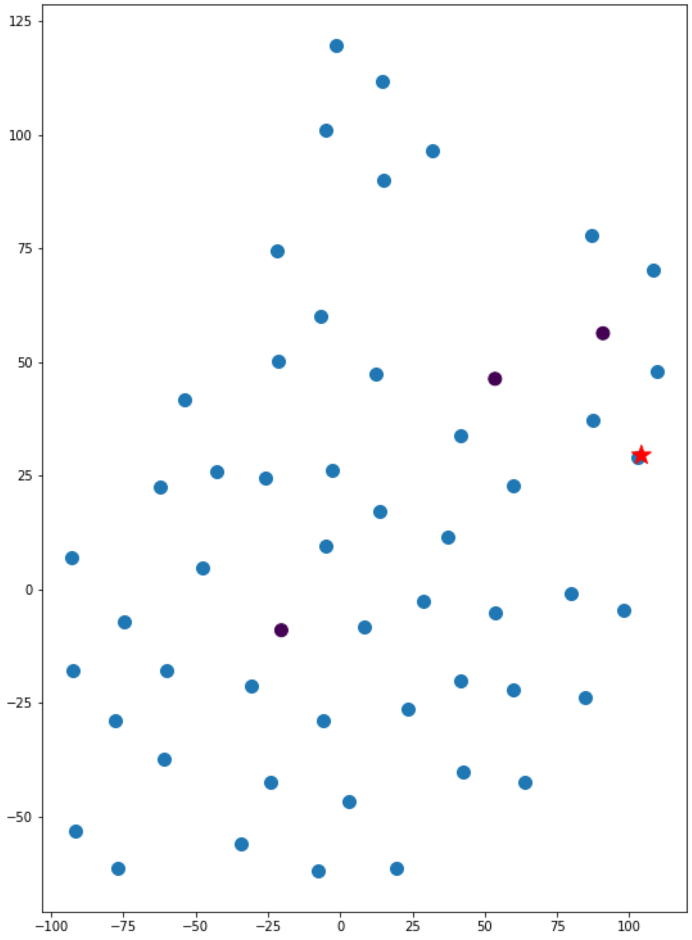

Figure 4: TSNE 2d feature space

4.2 Problem and Solution

Cluster: 2
Write a function that receives as argument a number and returns the factorial of the given number.
Suppose the following input is supplied to the program:
8
Then, the output should be:
40320

```
def fatorial(number):
    total = 1

    for i in range(number, 1, -1):
        total = total * i

    return total
```

5 Problem with solution 8

5.1 TSNE representation

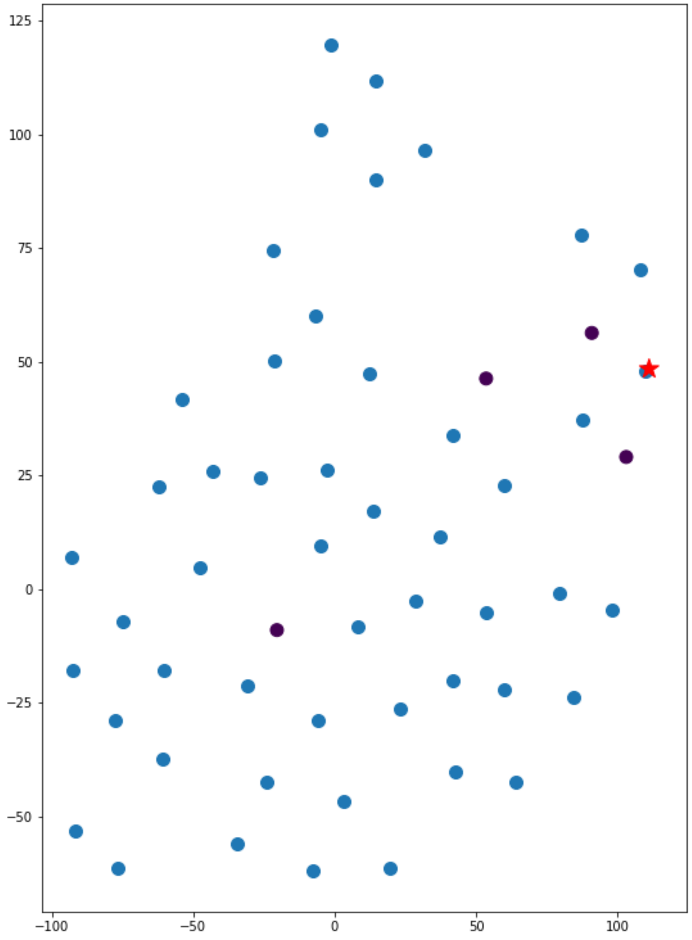

Figure 5: TSNE 2d feature space

5.2 Problem and Solution

Cluster: 2
Create a function that gets a number as an argument and returns a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)

```
def divisors(num):
    listRange = list(range(1,num+1))
    divisorList = []

    for number in listRange:
        if num % number == 0:
            divisorList.append(number)

    return divisorList
```

6 Problem with solution 2

6.1 TSNE representation

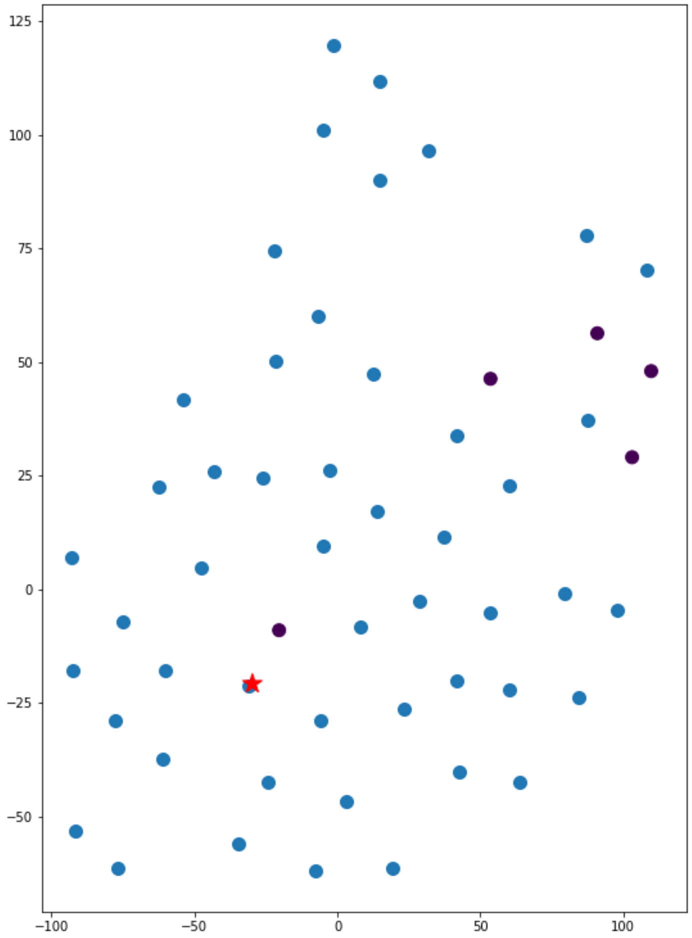

Figure 6: TSNE 2d feature space

6.2 Problem and Solution

Cluster: 2
Implement a function that takes as input three variables, and returns the largest of the three. Do this without using the Python max() function! The goal of this exercise is to think about some internals that Python normally takes care of for us. All you need is some variables and if statements!

```
def max_of_three(a,b,c):
     max_3=0
     if a>b:
         #max_3=a
         if a>c:
             max_3=a
         else:
             max_3=c
     else:
          if b>c:
             max_3=b
          else:
             max_3=c
     return max_3
```

7 Problem with solution 20

7.1 TSNE representation

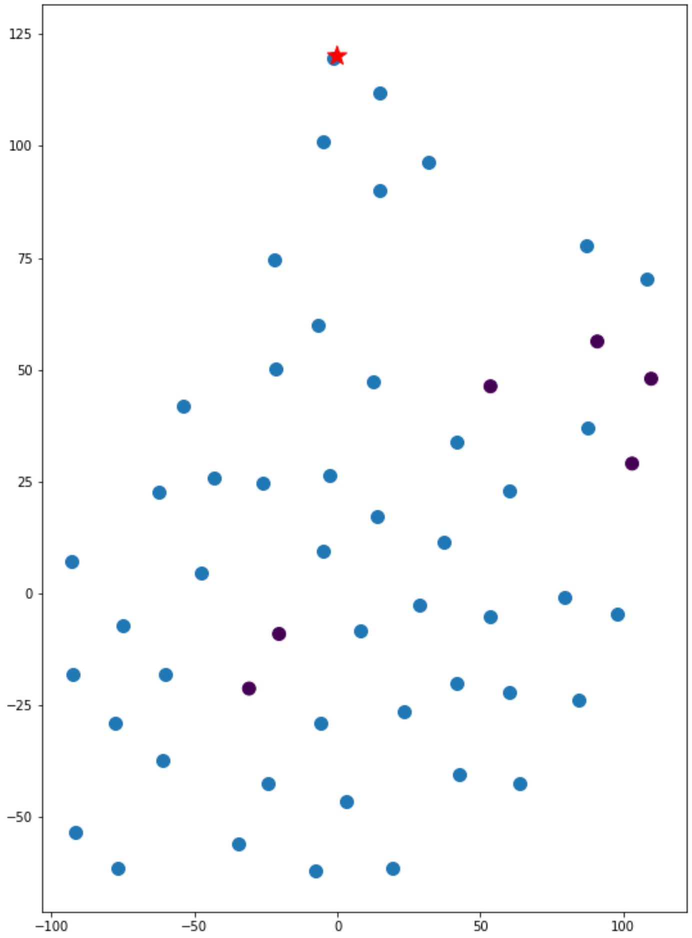

Figure 7: TSNE 2d feature space

7.2 Problem and Solution

Cluster: 0
Write a function to:
1) Convert a certain amount into Euros using the given exchange rate (you should divide the amount by the exchange rate).
2) Calculate how many 50, 20, 10, 5 and 1 Euro notes/coins you would receive for a given value of Euros.

You should ignore the cents after the conversion.

Your function should return, in a tuple, in this order: total amount in euros, the how many 50 euros notes, 20 notes, 10 notes, 5 notes and the remaining.

```
def euro_conversion(amount, exchange_rate):
    euro = int(amount//exchange_rate)

    euro50s = int(euro // 50)
    remainingEuros = euro % 50

    euro20s = int(remainingEuros // 20)
    remainingEuros = remainingEuros % 20

    euro10s = int(remainingEuros // 10)
    remainingEuros = remainingEuros % 10

    euro5s = int(remainingEuros // 5)
    remainingEuros = remainingEuros % 5

    return(euro, euro50s, euro20s, euro10s, euro5s, remainingEuros)
```

8 Problem with solution 48

8.1 TSNE representation

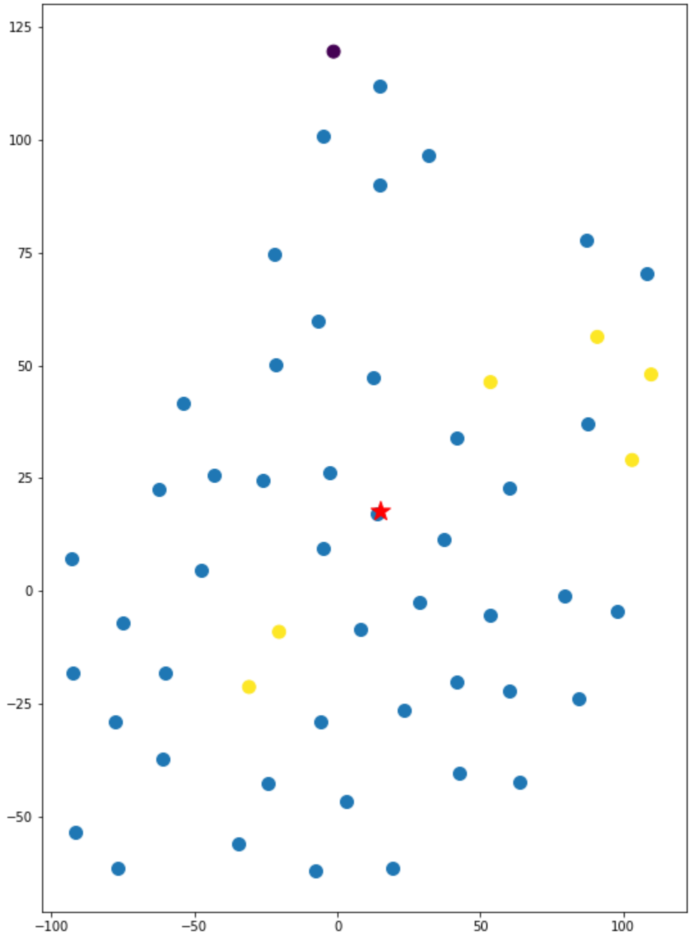

Figure 8: TSNE 2d feature space

8.2 Problem and Solution

Cluster: 2
Write a function that computes the net amount of a bank account based a transaction log passed as a list in the function argument. The transaction log format is shown as following:
[('D', 100), ('W', 200)] where D means deposit while W means withdrawal.
Suppose the following input is supplied to the program:
[('D', 300), ('D', 300), ('W', 200), ('D', 100)]

Then, the output should be:
500

```
def amount(transaction_list):
    total = 0
    for item in transaction_list:
        op, value = item
        if op == 'D':
            total += value
        elif op == 'W':
            total -= value
    return total
```

9 Problem with solution 46

9.1 TSNE representation

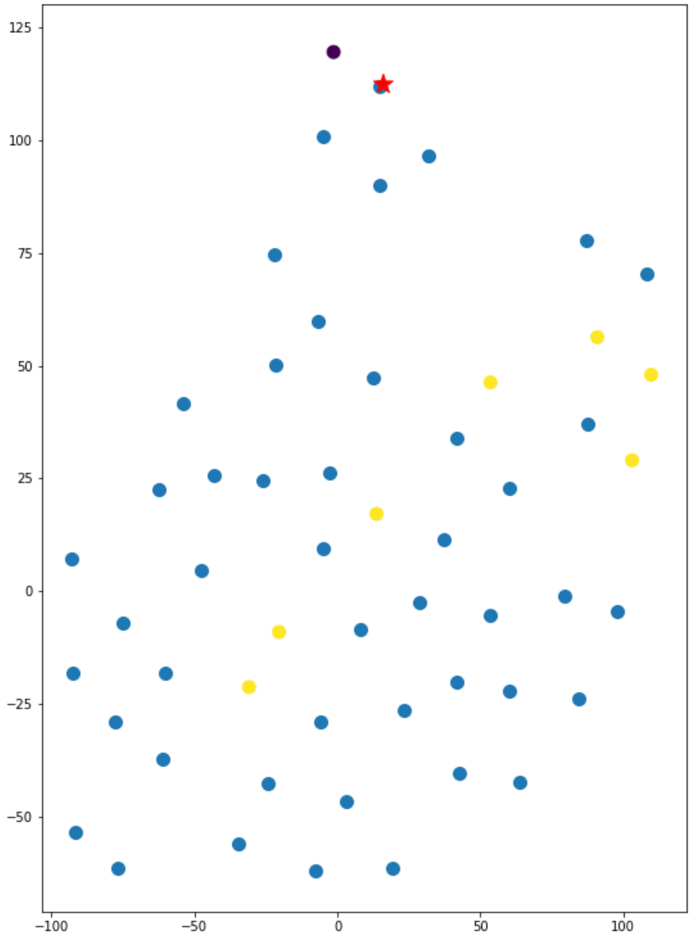

Figure 9: TSNE 2d feature space

9.2 Problem and Solution

Cluster: 0
Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.
Suppose the following input is supplied to the program:
9
Then, the output should be:
11106

```
def digit_sum(digit):
    n1 = int("{0}".format(digit))
    n2 = int("{0}{1}".format(digit,digit))
    n3 = int("{0}{1}{2}".format(digit,digit,digit))
    n4 = int("{0}{1}{2}{3}".format(digit,digit,digit,digit))
    total = n1+n2+n3+n4
    return total
```

10 Problem with solution 16

10.1 TSNE representation

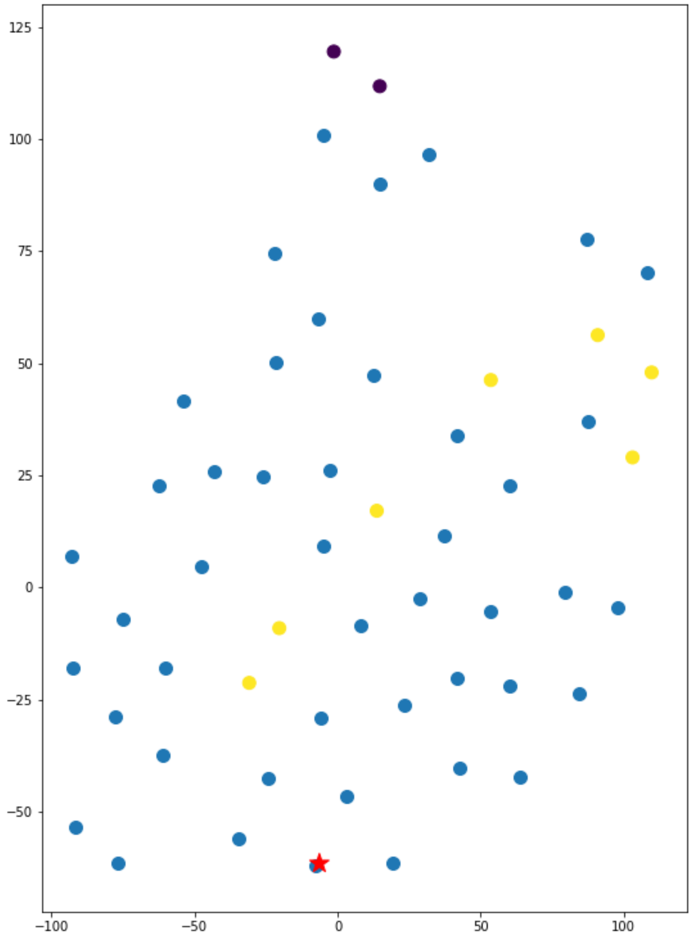

Figure 10: TSNE 2d feature space

10.2 Problem and Solution

Cluster: 2
Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns an appropriate boolean.

```
def find(ordered_list, element_to_find):
  for element in ordered_list:
    if element == element_to_find:
      return True
  return False
```

11 Problem with solution 14

11.1 TSNE representation

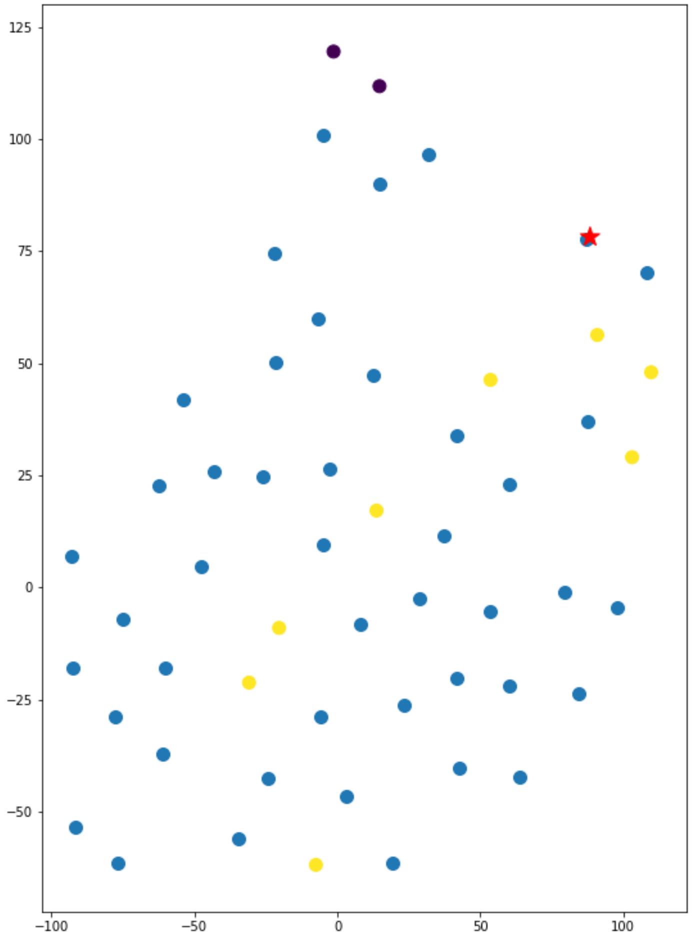

Figure 11: TSNE 2d feature space

11.2 Problem and Solution

Cluster: 1
Write a function that receives as an argument the number of how many Fibonnaci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. (Hint: The Fibonnaci seqence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, . . .)

```
def gen_fib(number):
    i = 1
    if number == 0:
        fib = []
    elif number == 1:
        fib = [1]
    elif number == 2:
        fib = [1,1]
    elif number > 2:
        fib = [1,1]
        while i < (number - 1):
            fib.append(fib[i] + fib[i-1])
            i += 1

    return fib
```

12 Problem with solution 44

12.1 TSNE representation

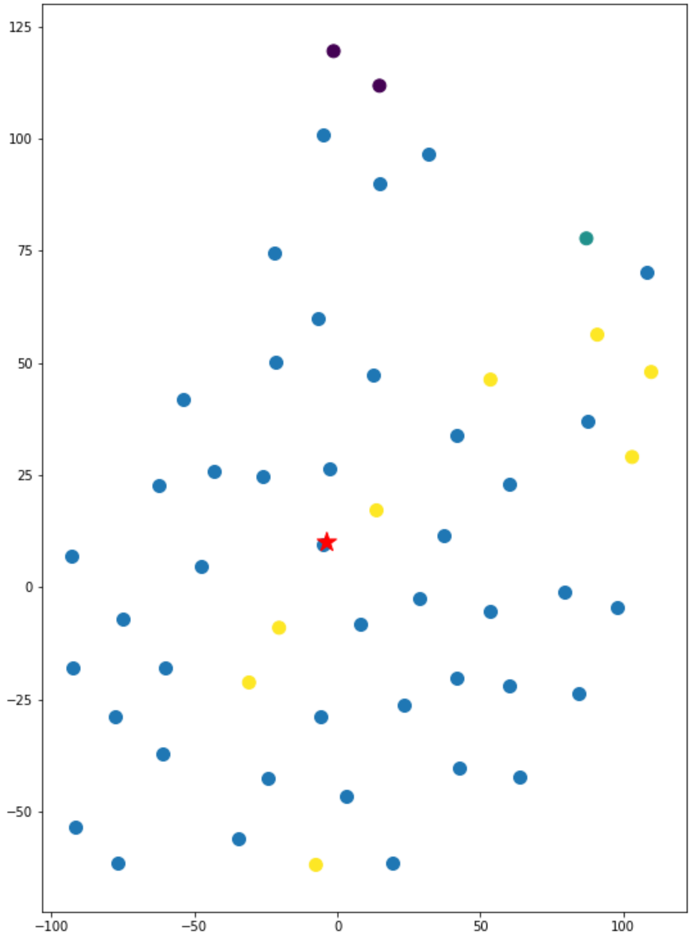

Figure 12: TSNE 2d feature space

12.2 Problem and Solution

Cluster: 2
Write a function that receives as argument a sentence and calculate the number of letters and digits.
Suppose the following input is supplied to the program:
hello world! 123
Then, the output should be the dictionary:
{"letters": 10, "digits": 3}

```
def count(sentence):
    d={"digits":0, "letters":0}
    for char in sentence:
        if char.isdigit():
            d["digits"]+=1
        elif char.isalpha():
            d["letters"]+=1
        else:
            pass
    return d
```

13 Problem with solution 30

13.1 TSNE representation

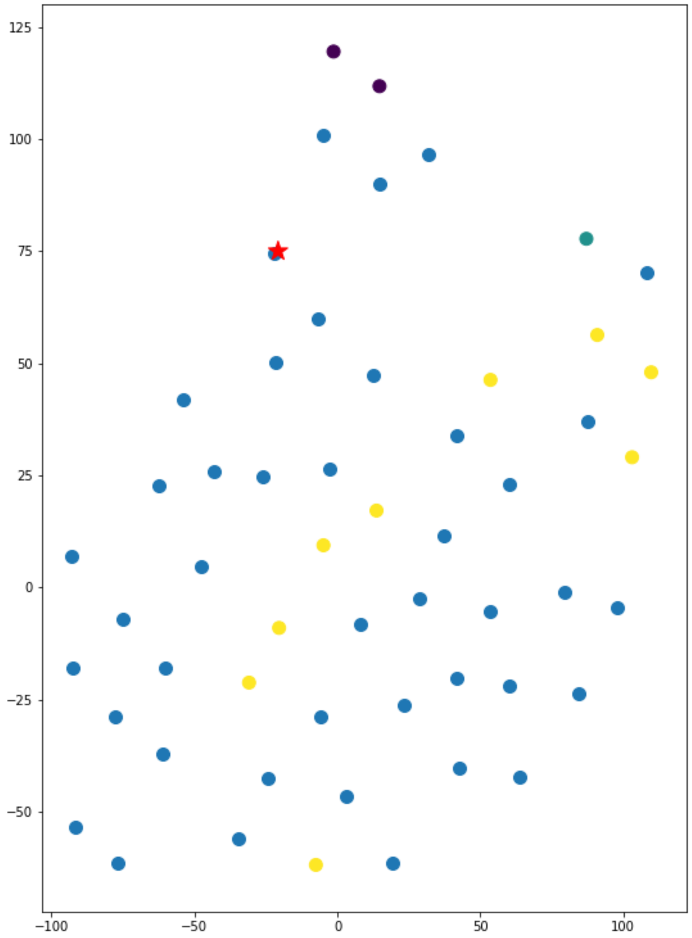

Figure 13: TSNE 2d feature space

13.2 Problem and Solution

Cluster: 1
Create a function that receives as argument a binary string of 8-bits and convert it into hexadecimal string equivalent.
To convert from binary to hexadecimal, divide the string into two parts of 4 bits and calculate the decimal equivalent.
If it's less than 10, just report the found number, if it's more than 10, convert it to the appropriate letter:
a) 10 = A
b) 11 = B
c) 12 = C
d) 13 = D
e) 14 = E
f) 15 = F

For example, the 63 decimal number is represented in binary as 00111111.
If we separate it in two parts, we get:
1) 0011
2) 1111

Transforming each part to its decimal equivalent, we get:
1) 3
2) 15

Then, using the code above, we get for each part:
1) 3
2) F

So, the hexadecimal number equivalent to the binary 00111111 is 3F.

```
def bin2hex(binaryString):
    hexA = binaryString[:4]
    hexB = binaryString[4:]
    finalhex = ""
    for eachSegment in (hexA, hexB):
        denary = int(eachSegment[0])*8 + int(eachSegment[1])*4 + int(eachSegment[2])*2 + int(eachSegment[3])*1
        if denary == 10:
            denary = 'A'
        elif denary == 11:
            denary = 'B'
        elif denary == 12:
            denary = 'C'
        elif denary == 13:
            denary = 'D'
        elif denary == 14:
            denary = 'E'
        elif denary == 15:
            denary = 'F'

        finalhex = finalhex + str(denary)
    return finalhex
```

14 Problem with solution 5

14.1 TSNE representation

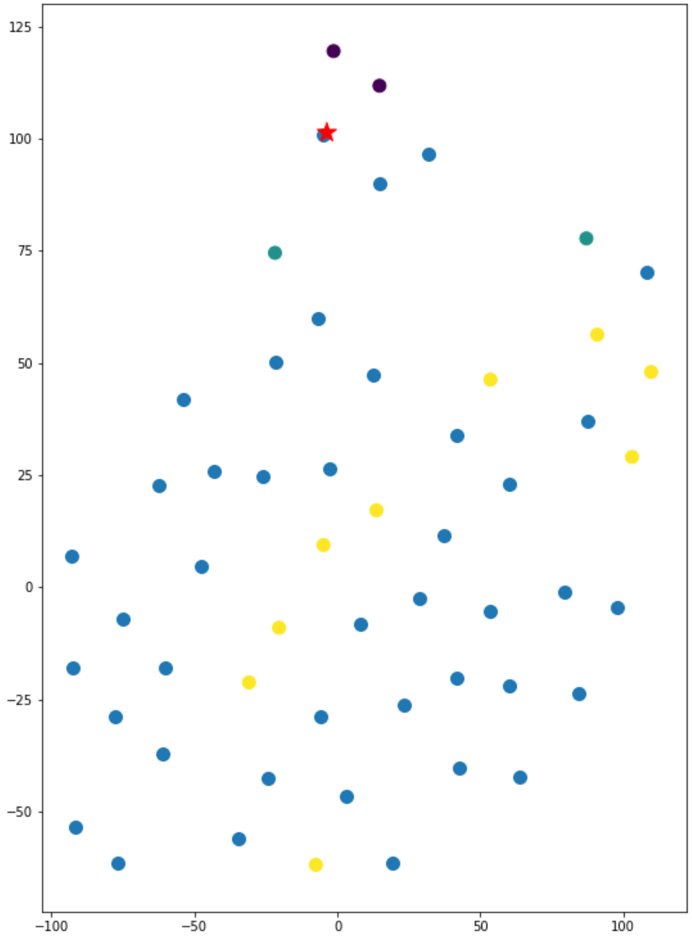

Figure 14: TSNE 2d feature space

14.2 Problem and Solution

Cluster: 0
Define a function that can receive two integral numbers in string form and compute their sum. Return the result as integer.

```
def sum_str(s1,s2):
return int(s1)+int(s2)
```

15 Problem with solution 45

15.1 TSNE representation

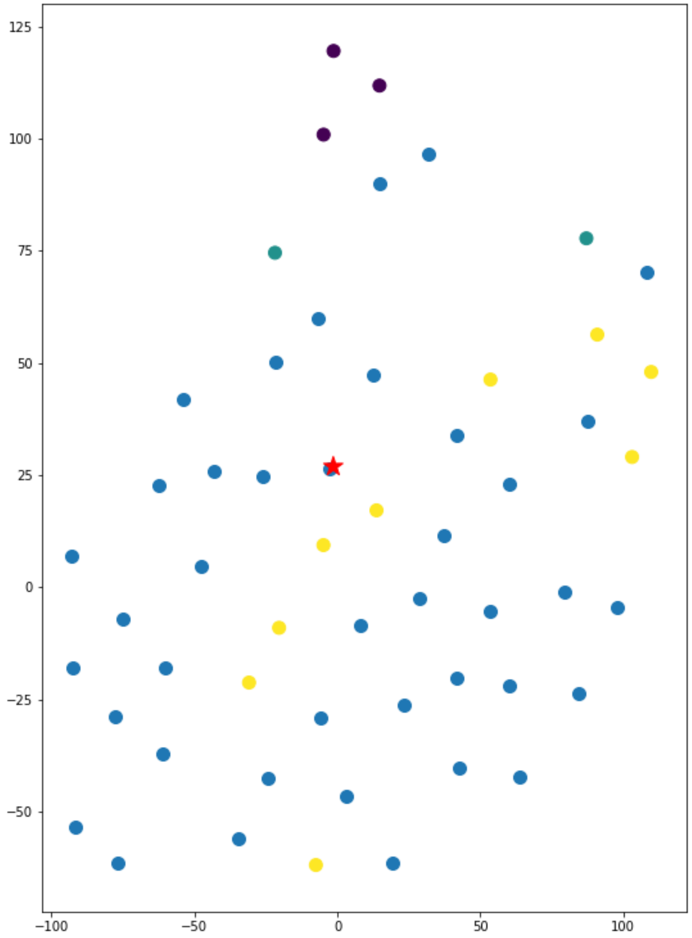

Figure 15: TSNE 2d feature space

15.2 Problem and Solution

Cluster: 2
Write a function that receives as argument a sentence and calculate the number of upper case letters and lower case letters.
Suppose the following input is supplied to the program:
Hello world!
Then, the output should be the dictionary:
{"uppercase": 1, "lowercase": 9}

```
def count(sentence):
    d={"uppercase":0, "lowercase":0}
    for char in sentence:
        if char.isupper():
            d["uppercase"]+=1
        elif char.islower():
            d["lowercase"]+=1
        else:
            pass
    return d
```

16 Problem with solution 1

16.1 TSNE representation

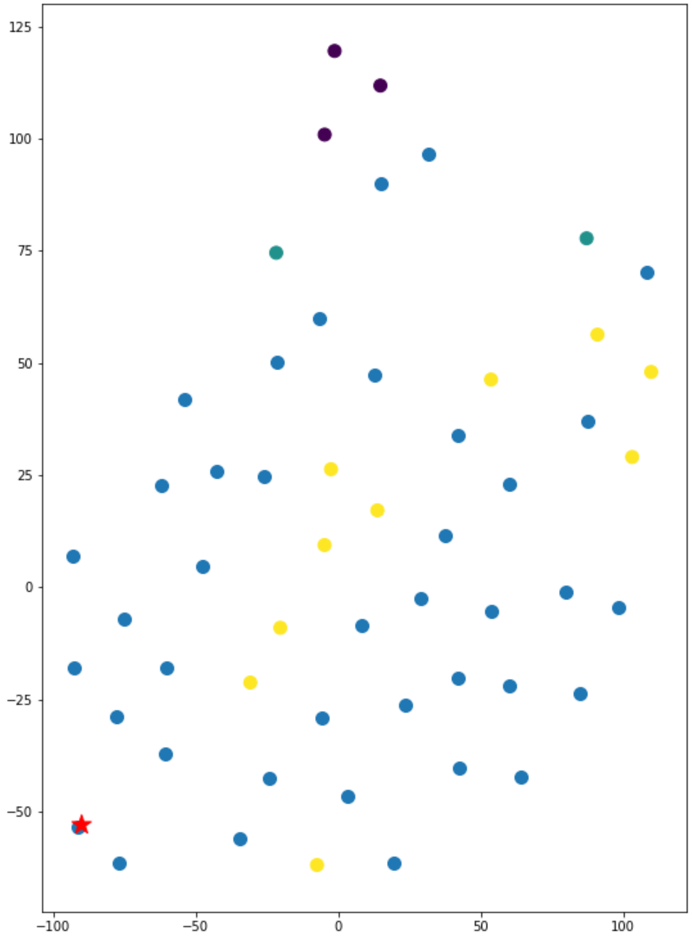

Figure 16: TSNE 2d feature space

16.2 Problem and Solution

Cluster: 0
Write a program (using functions!) that receives as an argument a long string containing multiple words. The function should return the same string, except with the words in backwards order. For example, say I type the string:
My name is Michele
Then I would see the string:
Michele is name My
shown back to me.

```
def reverseWord(w):
  return ' '.join(w.split()[::-1])
```

17 Problem with solution 49

17.1 TSNE representation

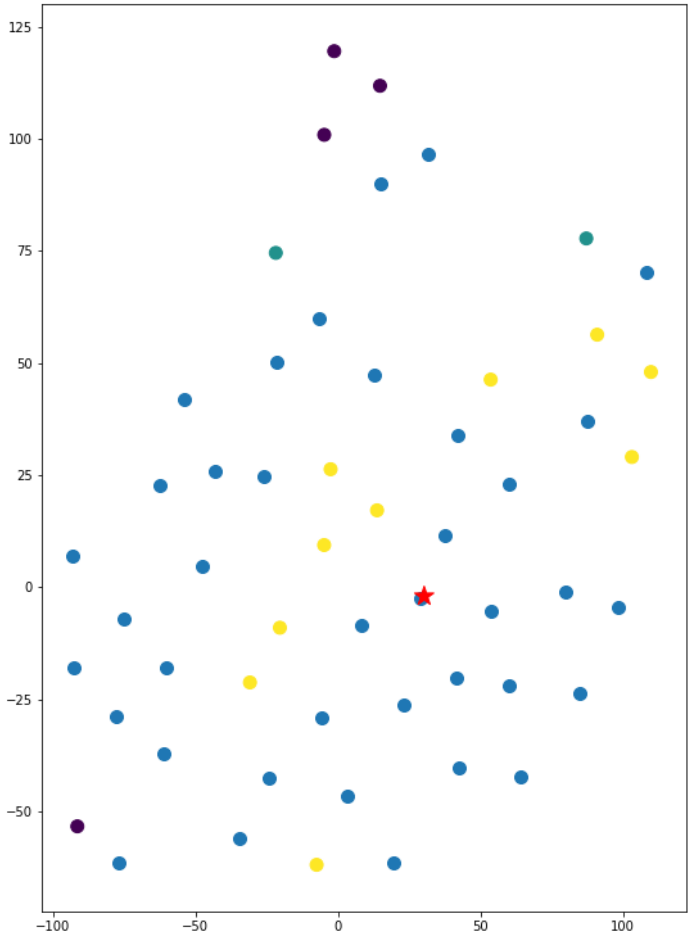

Figure 17: TSNE 2d feature space

17.2 Problem and Solution

Cluster: 2
Write a function that computes the net amount of a bank account based a transaction log passed as a list in the function argument. The transaction log format is shown as following:
[('D', 100), ('W', 200)] where D means deposit while W means withdrawal.
Suppose the following input is supplied to the program:
[('D', 300), ('D', 300), ('W', 200), ('D', 100)]

Then, the output should be:
500

```
# Complete your function using this header
def amount(transaction_list):
    op = {"D": 1, "W": -1}
    values = [op[item[0]]*item[1] for item in transaction_list]
    return sum(values)
```

18 Problem with solution 31

18.1 TSNE representation

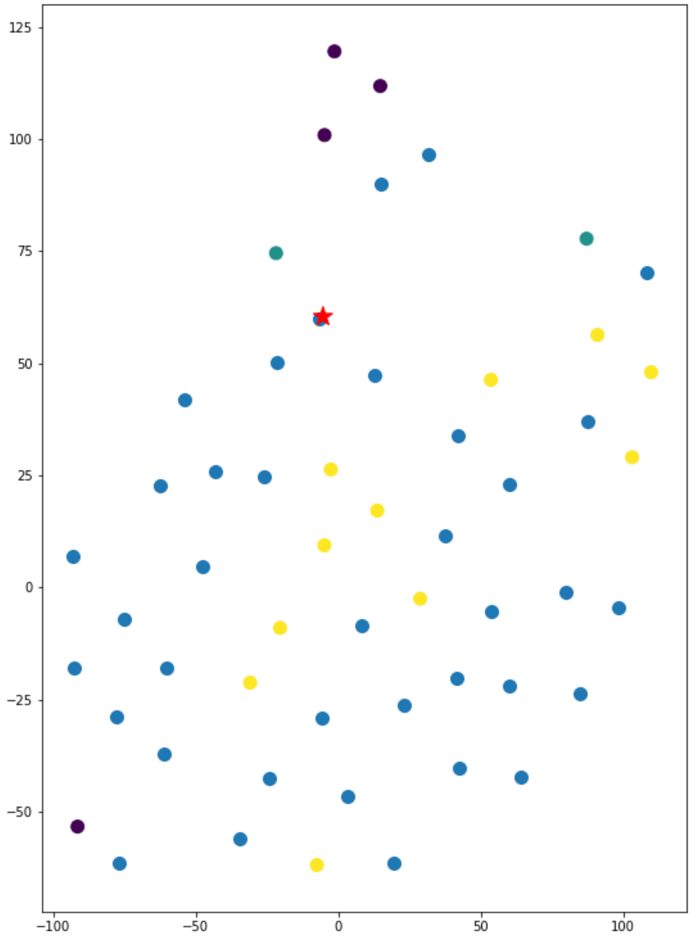

Figure 18: TSNE 2d feature space

18.2 Problem and Solution

Cluster: 1
Create a function that receives as argument a hexadecimal string and convert it into its decimal equivalent number.
Do not forget, that the hexadecimal string can vary in length.

```
def hex2dec(hexString):
    denary = 0
    lengthHex = len(hexString)
    for element in range(lengthHex):
        hexSeg = hexString[element]
        if hexSeg == 'A':
            hexSeg = 10
        elif hexSeg == 'B':
            hexSeg = 11
        elif hexSeg == 'C':
            hexSeg = 12
        elif hexSeg == 'D':
            hexSeg = 13
        elif hexSeg == 'E':
            hexSeg = 14
        elif hexSeg == 'F':
            hexSeg = 15
        else:
            hexSeg = int(hexSeg)

        #work out the place value power of 16
        placePower = 16**(lengthHex-(element+1))
        hexSeg = hexSeg * placePower
        denary = denary + hexSeg

    return denary
```

19 Problem with solution 33

19.1 TSNE representation

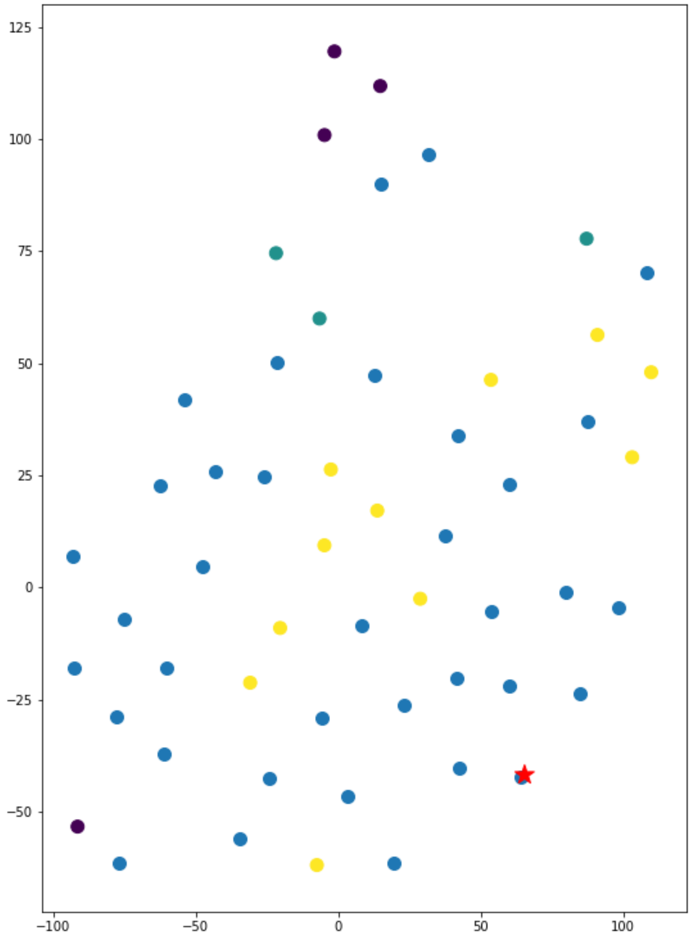

Figure 19: TSNE 2d feature space

19.2 Problem and Solution

Cluster: 2
Write a function which will find all such numbers which are divisible by 7 but are not a multiple of 5,
between 2000 and 3200 (both included).
The numbers obtained should be returned in a list

```
def divisible():
    l=[]
    for i in range(2000, 3201):
        if (i%7==0) and (i%5!=0):
            l.append(i)
    return l
```

20 Problem with solution 39

20.1 TSNE representation

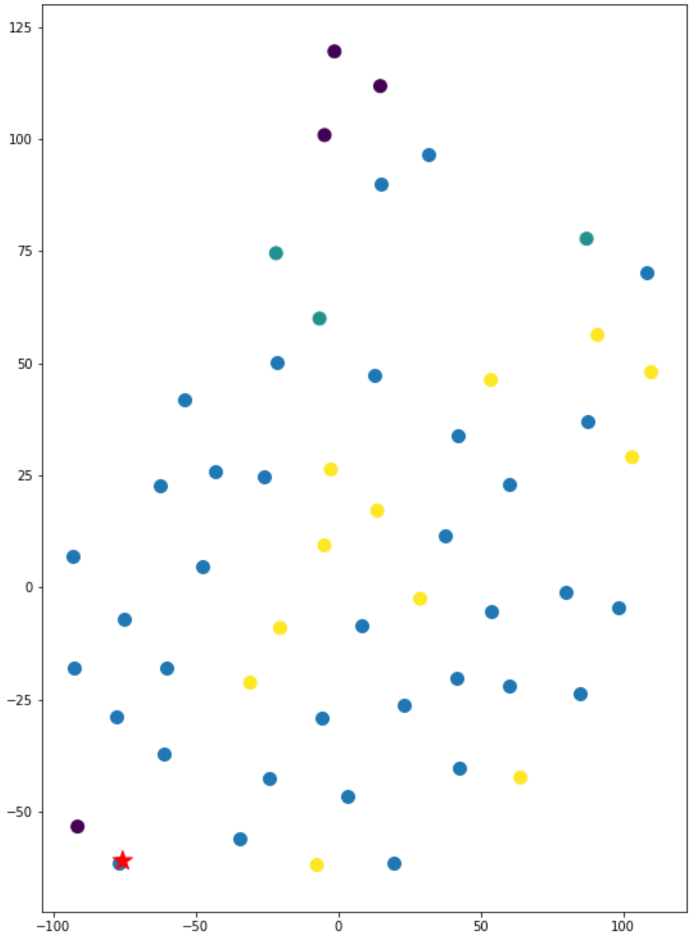

Figure 20: TSNE 2d feature space

20.2 Problem and Solution

Cluster: 0
Write a function that receives a comma separated sequence of words as argument and returns the words in a comma-separated sequence after sorting them alphabetically.
Suppose the following input is supplied to the program:
without, hello, bag, world
Then, the output should be:
bag, hello, without, world

Pay attention to the spaces after the commas!

```
def sort_csv(csv):
    items = csv.split(’, ’)
    items.sort()
    return ", ".join(items)
```

21 Problem with solution 21

21.1 TSNE representation

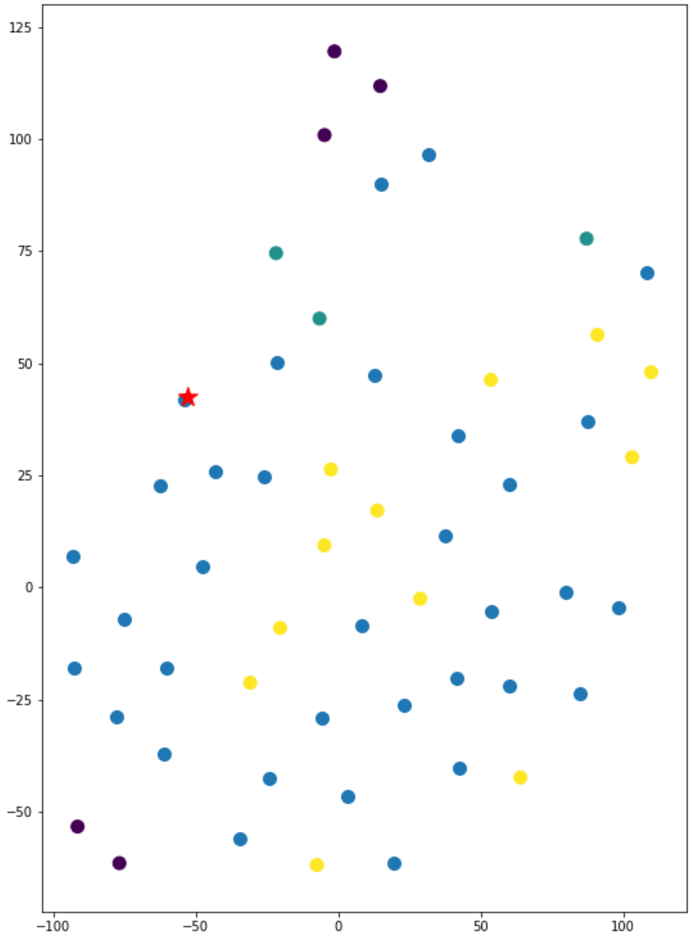

Figure 21: TSNE 2d feature space

21.2 Problem and Solution

Cluster: 1
Write a function that receives as argument a month (from 1 to 12) and returns its season. Remembering:
a) December, January and February are summer
b) March, April and May are autumn
c) June, July and August are winter
d) September, October and November are spring

```
def season(month):
    if month == 1 or month == 2 or month == 12:
        return "summer"
    elif month == 3 or month == 4 or month == 5:
        return "autumn"
    elif month == 6 or month == 7 or month == 8:
        return "winter"
    elif month == 9 or month == 10 or month == 11:
        return "spring"
```

22 Problem with solution 52

22.1 TSNE representation

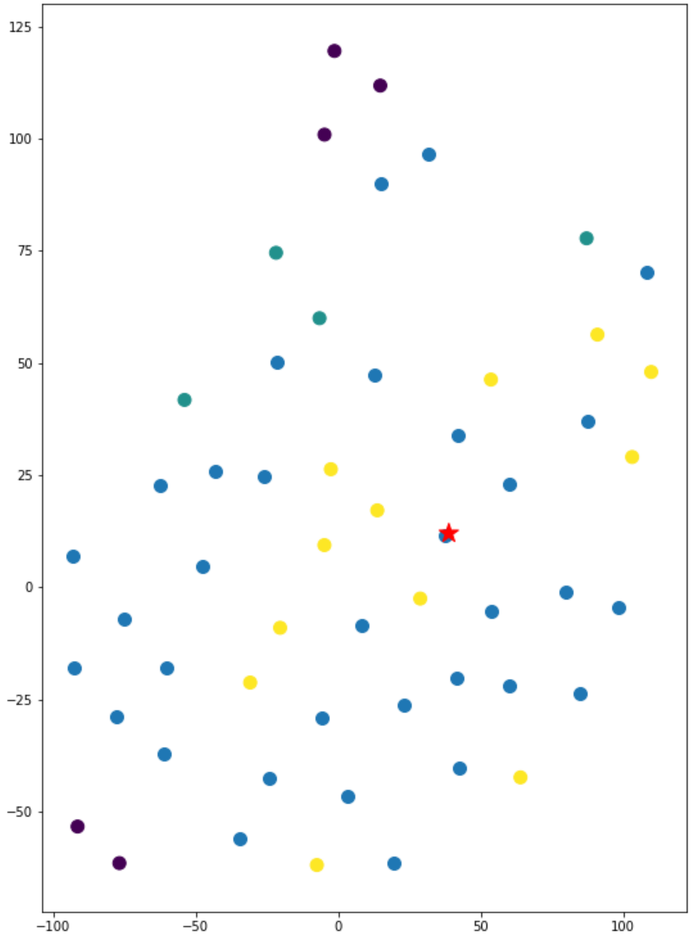

Figure 22: TSNE 2d feature space

22.2 Problem and Solution

Cluster: 2
Write a program to compute the frequency of the words in a sentence. The output should be a dictionary where the key are the words and the value the respective frequency in the sentence.
Do not worry about punctuation.
Suppose the following input is supplied to the program:
New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.
Then, the output should be:
{"2":2, "3.":1, "3?":1, "New":1, "Python":5, "Read":1, "and":1, "between":1, "choosing":1, "or":2}

```
def word_freq(sentence):
    freq = {}   # frequency of words in text
    for word in sentence.split():
        freq[word] = freq.get(word,0)+1

    return freq
```

23 Problem with solution 0

23.1 TSNE representation

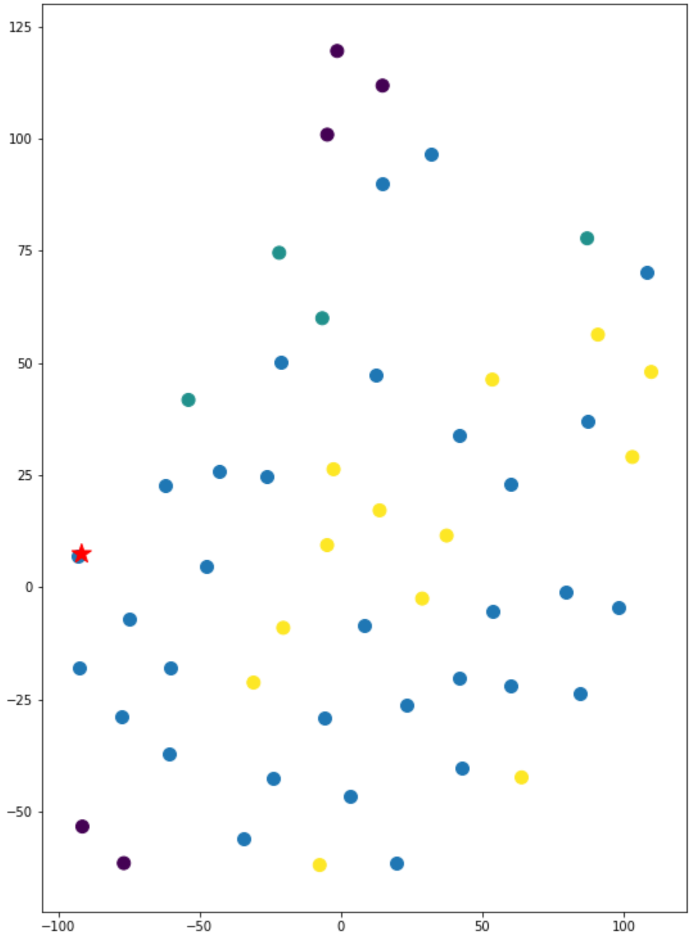

Figure 23: TSNE 2d feature space

23.2 Problem and Solution

Cluster: 0

Write a program that takes a list of numbers (for example, a = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list. For practice, write this code inside a function.

```
def list_ends(a_list):
    return [a_list[0], a_list[len(a_list)-1]]
```

24 Problem with solution 24

24.1 TSNE representation

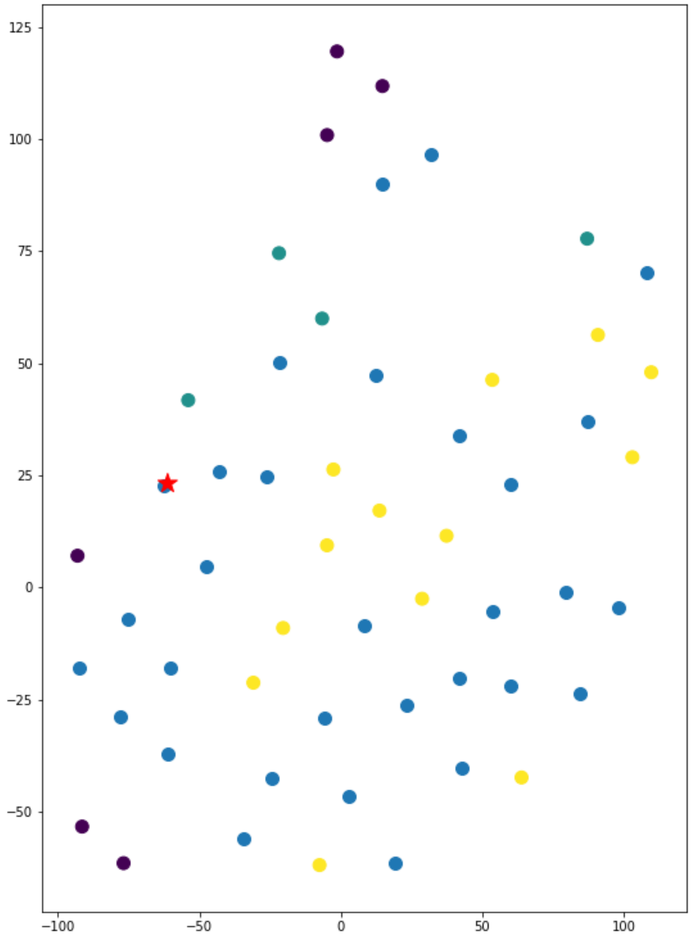

Figure 24: TSNE 2d feature space

24.2 Problem and Solution

Cluster: 1
Create a function that will output the grade according to the exam score.
Use the following rules:
1) If the user got 80 points or more, the function should return the grade A.
2) If the user got between 60 and 79, they got a B.
3) If the user got between 40 and 59, they got a C.
4) If the user got between 20 and 39, they got a D.
5) If the user got less then 20 points, they get an E.

```
def grade(score):
    if score >= 80:
        return 'A'
    elif score >= 60:
        return 'B'
    elif score >= 40:
        return 'C'
    elif score >= 20:
        return 'D'
    else:
        return 'E'
```

25 Problem with solution 10

25.1 TSNE representation

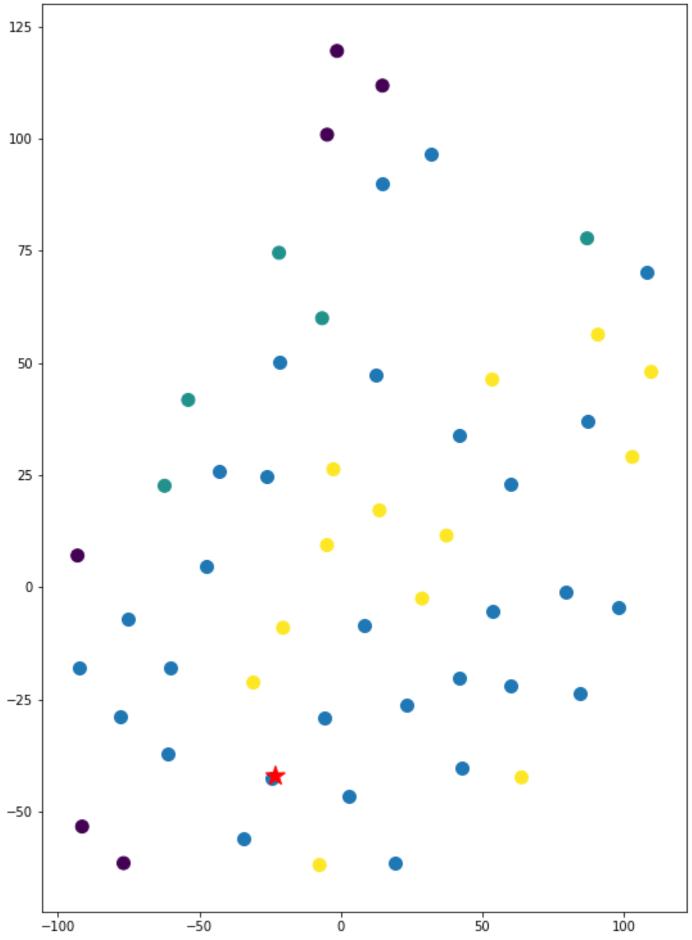

Figure 25: TSNE 2d feature space

25.2 Problem and Solution

Cluster: 2
Write a function that takes as argument a word and returns (boolean type True or False) whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

```
def palindrome(word):
    rvs = word[::-1]
    if word == rvs:
        return True
    else:
        return False
```

26 Problem with solution 3

26.1 TSNE representation

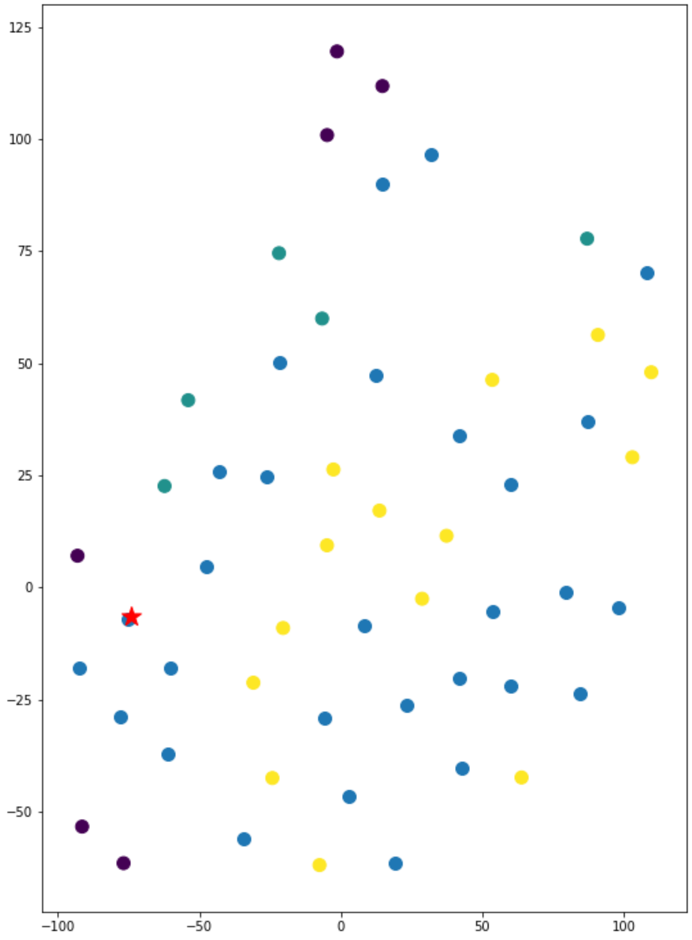

Figure 26: TSNE 2d feature space

26.2 Problem and Solution

Cluster: 0
Define a function which can compute the sum of two numbers.

```
def SumFunction(number1, number2):
return number1+number2
```

27 Problem with solution 12

27.1 TSNE representation

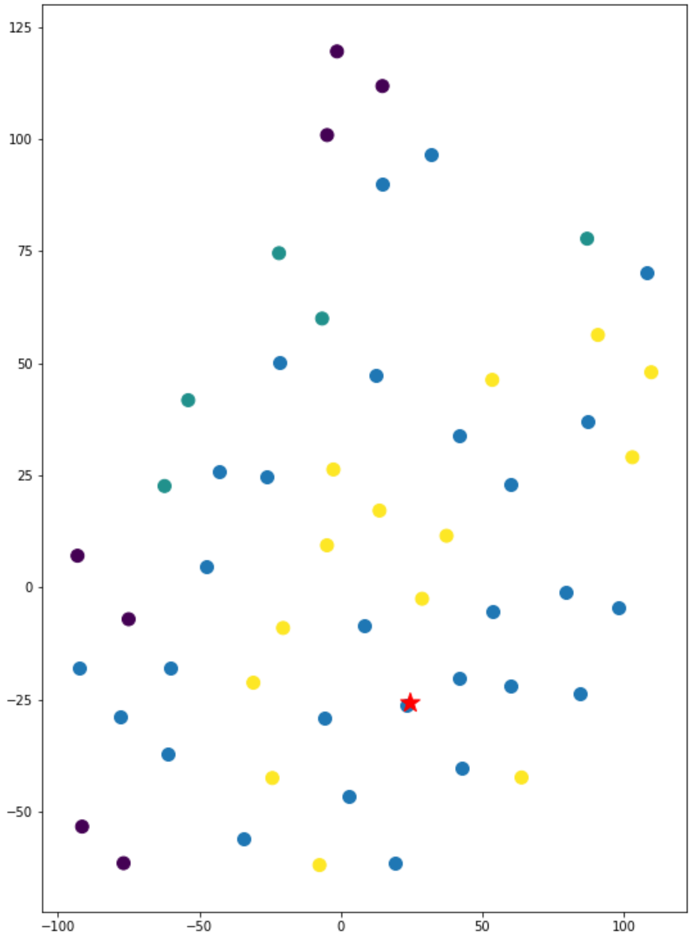

Figure 27: TSNE 2d feature space

27.2 Problem and Solution

Cluster: 2
Let's say I give you a list as an argument in a function (for example: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]). Write one Python function that takes this list as an argument and returns a new list that has only the even elements of this list in it.

```
def even(numlist):
    return [item for item in numlist if not item % 2]
```

28 Problem with solution 18

28.1 TSNE representation

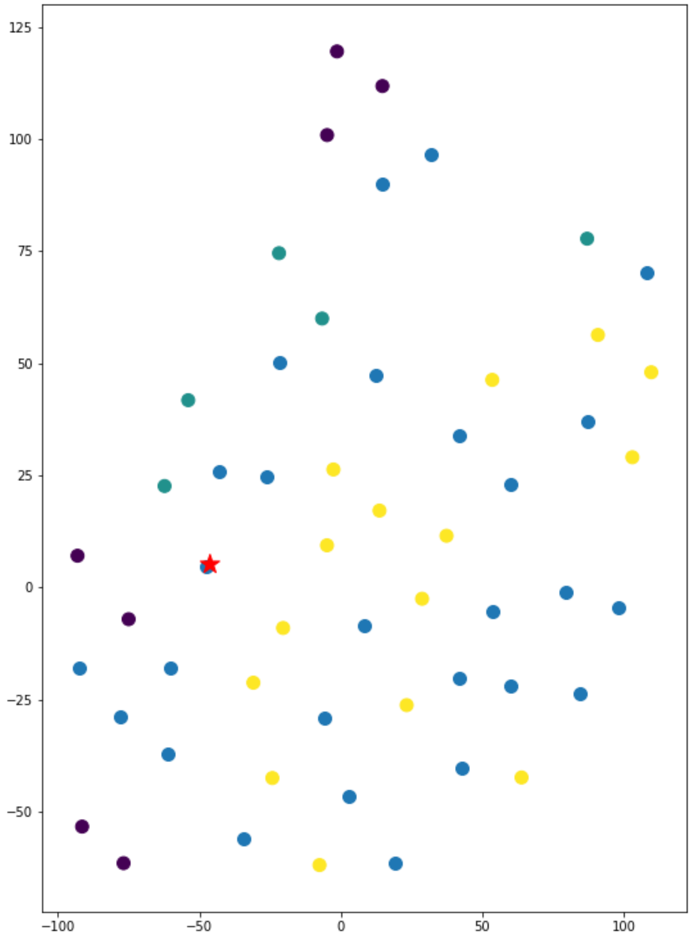

Figure 28: TSNE 2d feature space

28.2 Problem and Solution

Cluster: 1
Write a function that receives as argument how many goals Team One scored and how many goals Team Two scored. Your function should return "Team One scores 3 points and Team Two scores 0 points." if Team One scored more goals than Team Two and vice-versa. If the number of goals is the same, the returned sentence should be "Both Team One and Team Two score 1 point.".

```
def score(team_one, team_two):
    if team_one > team_two:
        return "Team One scores 3 points and Team Two scores 0 points."
    elif team_one < team_two:
        return "Team Two scores 3 points and Team One scores 0 points."
    else:   #Game must be a draw if the previous two if statements are false
        return "Both Team One and Team Two score 1 point."
```

29 Problem with solution 4

29.1 TSNE representation

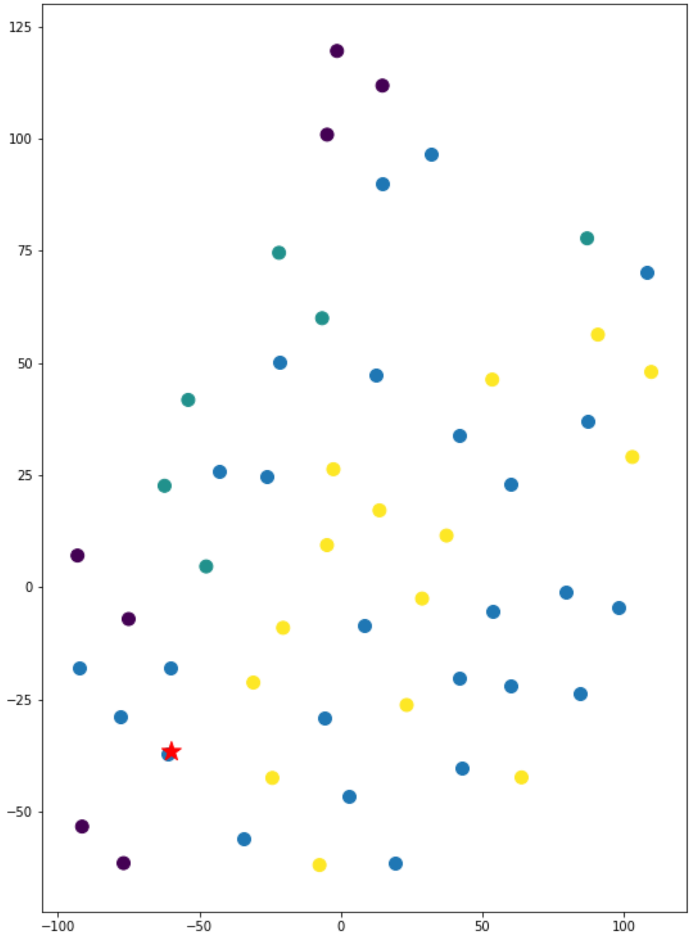

Figure 29: TSNE 2d feature space

29.2 Problem and Solution

Cluster: 0
Define a function that can convert a integer into a string and return the string.

```
def int2str(n):
return str(n)
```

30 Problem with solution 23

30.1 TSNE representation

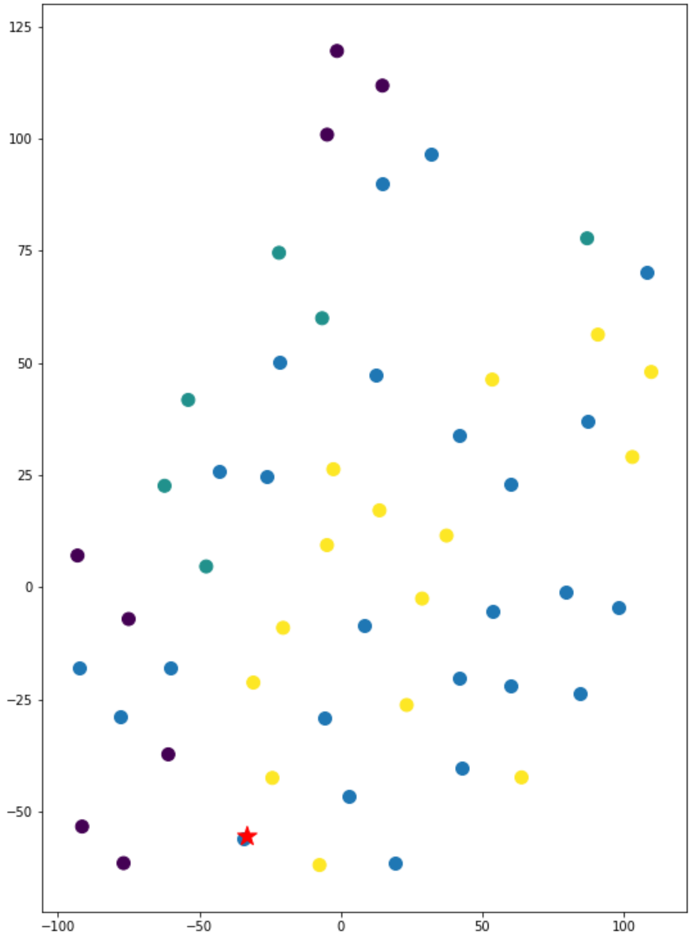

Figure 30: TSNE 2d feature space

30.2 Problem and Solution

Cluster: 2
Create a function that receives as argument the state of two switches (either 1 (on) or 0 (off)). The function should work out if both switches are on and then output True (the light is on). Otherwise, the program should output False (the light is off).

```
def light(switchA, switchB):
    if switchA == 1 and switchB == 1:
        return True
    else:
        return False
```

31 Problem with solution 6

31.1 TSNE representation

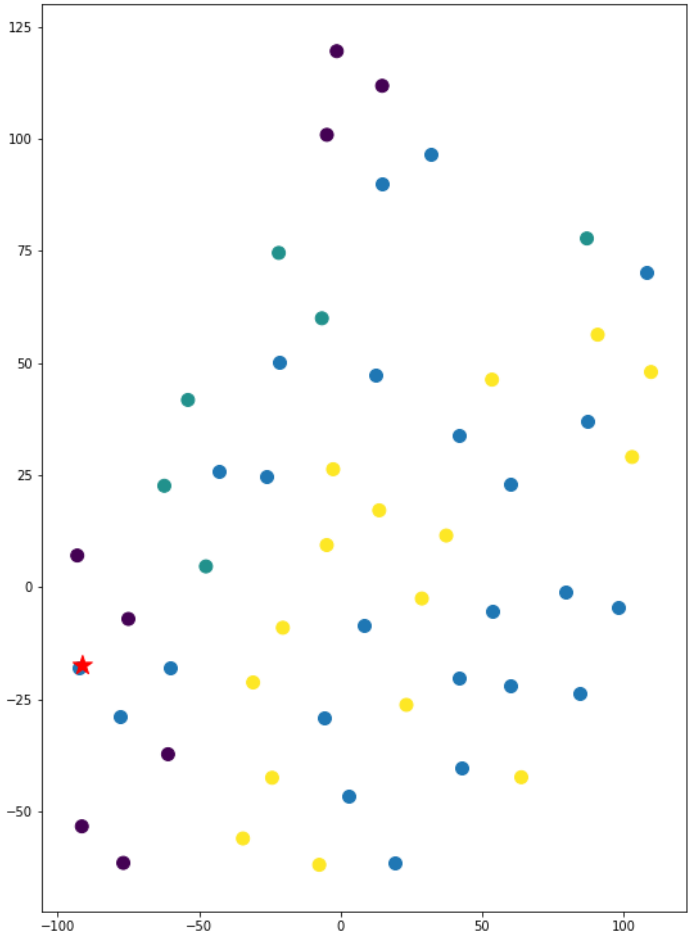

Figure 31: TSNE 2d feature space

31.2 Problem and Solution

Cluster: 0
Define a function that can accept two strings as input and concatenate them. Return the result.

```
def str_concat(s1,s2):
return s1+s2
```

32 Problem with solution 25

32.1 TSNE representation

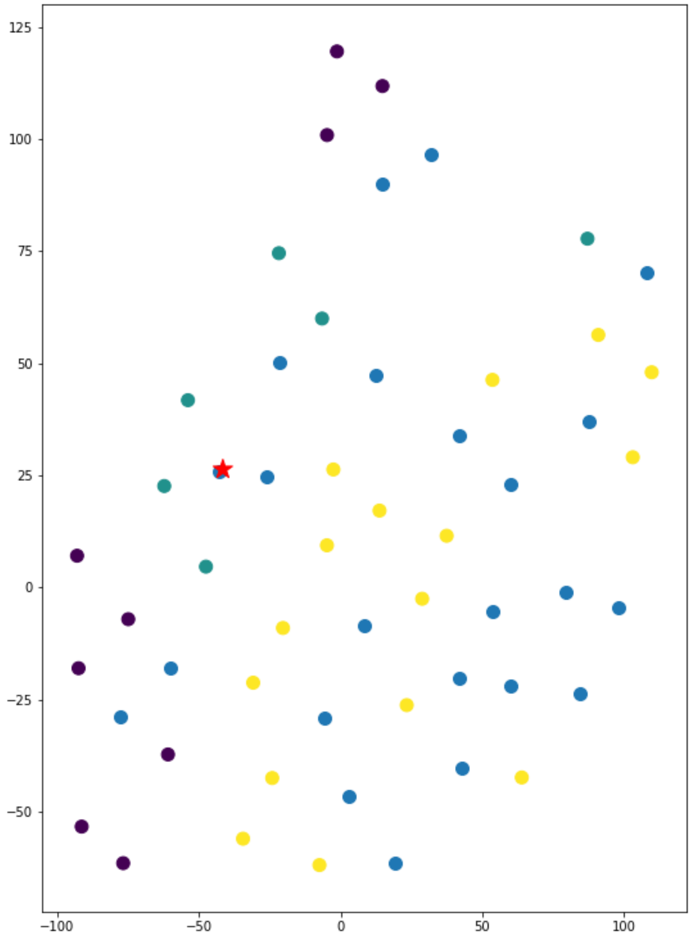

Figure 32: TSNE 2d feature space

32.2 Problem and Solution

Cluster: 1
Write a function that receives as argument a month (from 1 to 12) and a year (four digits) and returns the number of days in the given month (consider leap years!)
The rules for leap years are:
1) If the year is a multiple of 4, it is USUALLY a leap year (example: 1984).
2) However, if it is a multiple of 100 (even though is a multiple of 4), then it's not a leap year, breaking rule number 1 (example: 1900)
3) Except if it's a multiple of 400, then it is a leap year again, breaking rule number 2 (example: 2000)

```
def days_in_month(month, year):
    if month == 2:
        if (year % 400) == 0:
            return 29
        elif (year % 100) == 0:
            return 28
        elif (year % 4) == 0:
            return 29
        else:
            return 28
    elif month in (4,6,9,11):
        return 30
    elif month in (1,3,5,7,8,10, 12):
        return 31
```

33 Problem with solution 19

33.1 TSNE representation

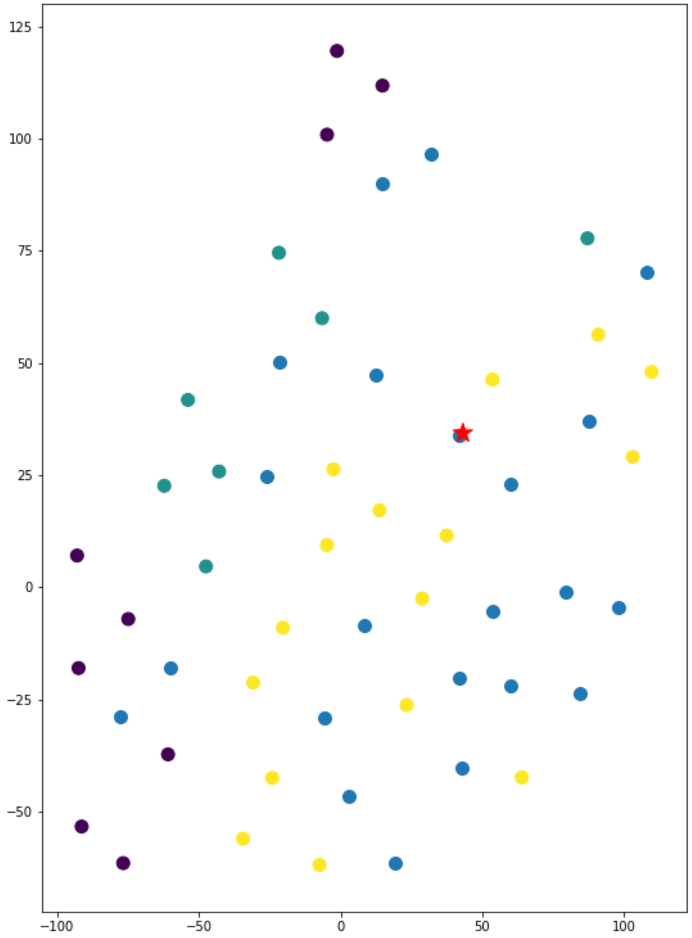

Figure 33: TSNE 2d feature space

33.2 Problem and Solution

Cluster: 2
Write a function that receives as an argument a list of numbers and returns the average (mean) of a these numbers.

```
def averaging(number_list):
    runningTotal = 0
    numberOfnumbers = len(number_list)
    # don’t forget to initialise the running total before you start
    for count in range(numberOfnumbers):
        nextNumber = number_list[count]
        runningTotal = runningTotal + nextNumber
    average = runningTotal/numberOfnumbers
    return average
```

34 Problem with solution 53

34.1 TSNE representation

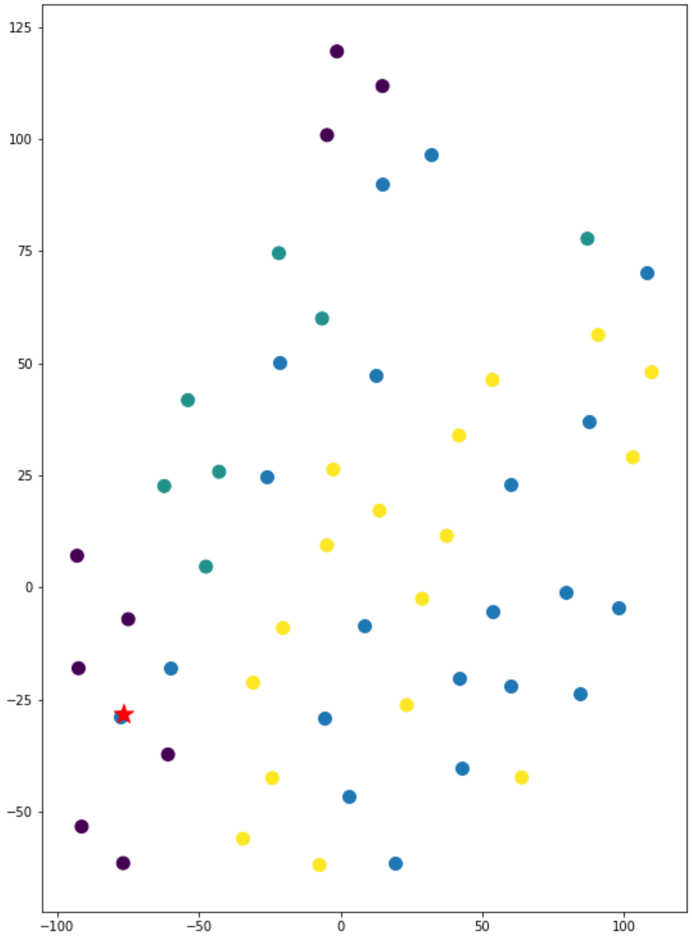

Figure 34: TSNE 2d feature space

34.2 Problem and Solution

Cluster: 0
Write a method which can calculate square value of number

```
def square(num):
    return num ** 2
```

35 Problem with solution 51

35.1 TSNE representation

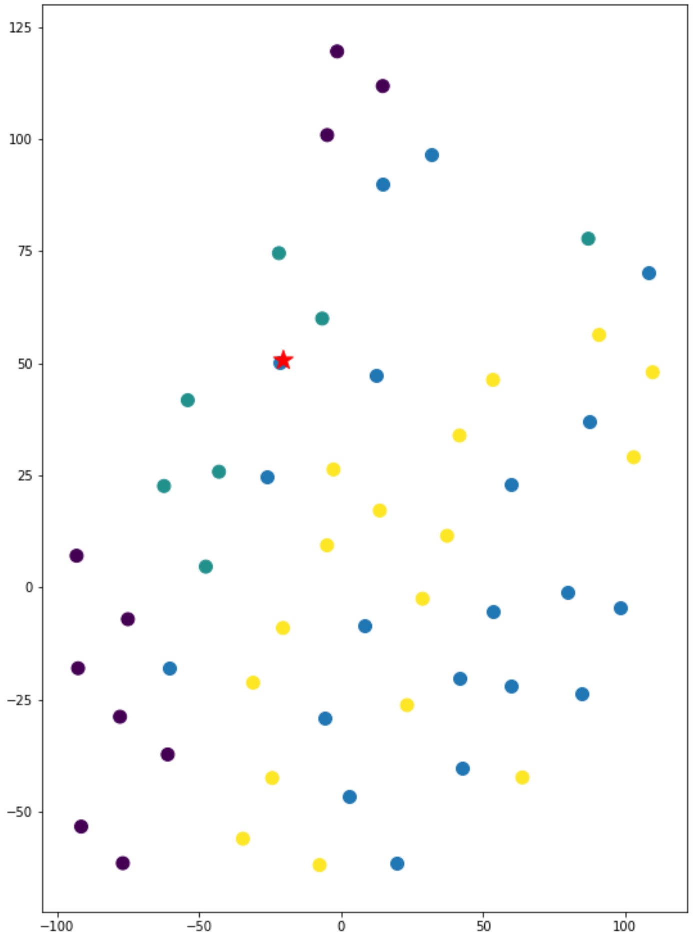

Figure 35: TSNE 2d feature space

35.2 Problem and Solution

Cluster: 1
A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps. The trace of robot movement is shown as the following:
[("UP", 5), ("DOWN", 3), ("LEFT", 3), ("RIGHT", 2)]

The numbers after the direction are steps. Please write a function to compute the distance from current position after a sequence of movement and original point. If the distance is a float, then just return the nearest integer. Make sure that the type you are returning is an int.
Example:
If the following tuples are given as input to the program:
[("UP", 5), ("DOWN", 3), ("LEFT", 3), ("RIGHT", 2)]
Then, the output of the program should be:
2

```
def robot_position(steps_list):
    import math
    pos = [0,0]

    for movement in steps_list:
        direction = movement[0]
        steps = int(movement[1])
        if direction=="UP":
            pos[0]+=steps
        elif direction=="DOWN":
            pos[0]-=steps
        elif direction=="LEFT":
            pos[1]-=steps
        elif direction=="RIGHT":
            pos[1]+=steps
        else:
            pass

    return round(math.sqrt(pos[1]**2+pos[0]**2))
```

36 Problem with solution 7

36.1 TSNE representation

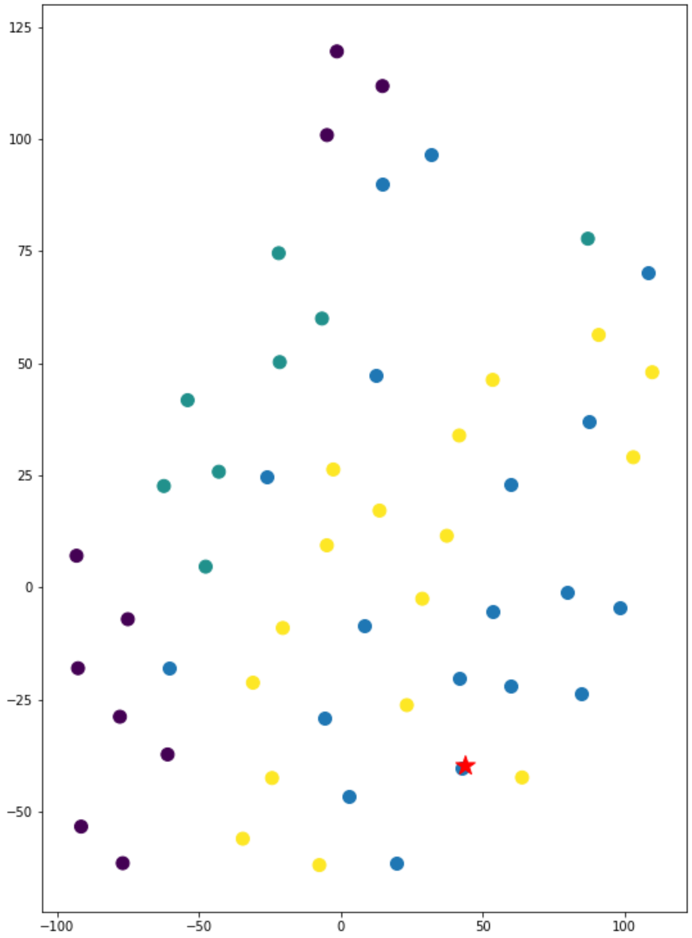

Figure 36: TSNE 2d feature space

36.2 Problem and Solution

Cluster: 2

Write a function that takes as argument a list (for example: numlist = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]) and returns only the elements of the list that are less than 5.

```
def lessthan5(num_list):
    new_list = []
    for item in num_list:
        if item < 5:
            new_list.append(item)

    return new_list
```

37 Problem with solution 42

37.1 TSNE representation

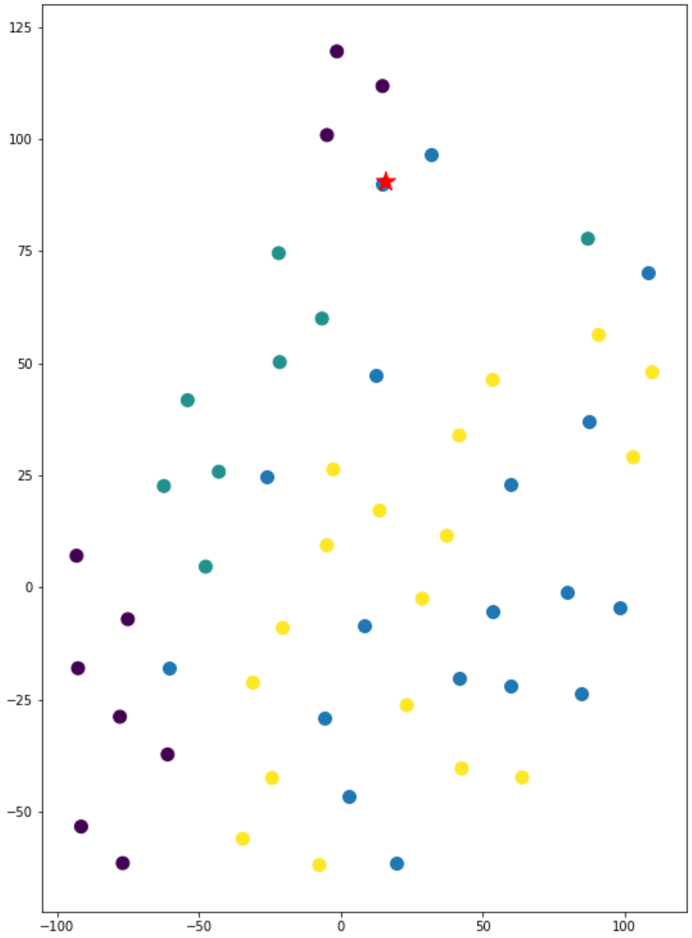

Figure 37: TSNE 2d feature space

37.2 Problem and Solution

Cluster: 0
Write a function, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be returned in a comma-separated sequence on a single line.

```
def all_even():
    values = []
    for i in range(1000, 3001):
        s = str(i)
        if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])
            values.append(s)
    return ", ".join(values)
```

38 Problem with solution 22

38.1 TSNE representation

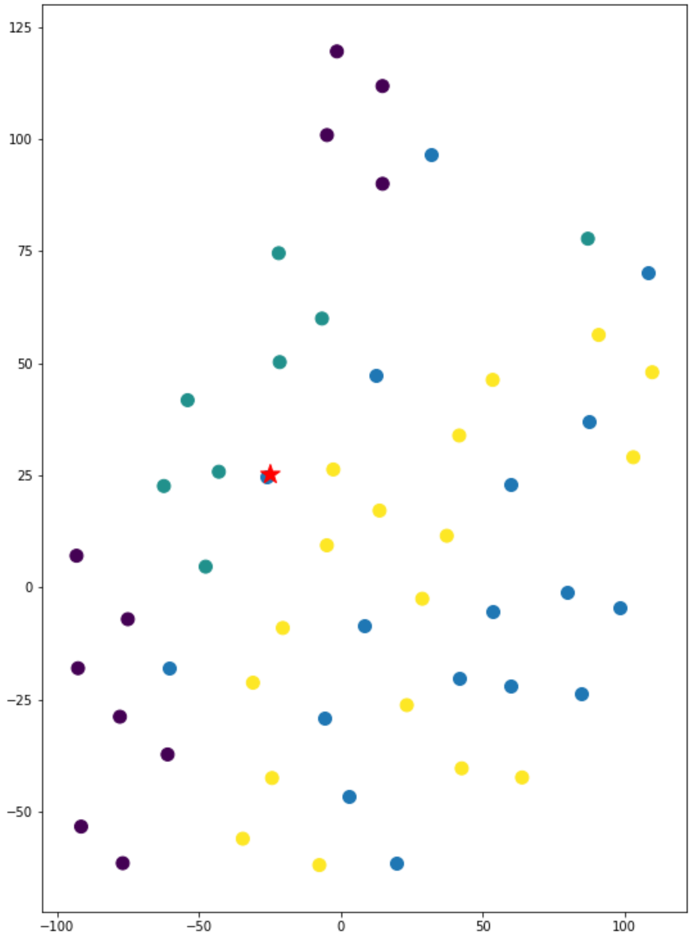

Figure 38: TSNE 2d feature space

38.2 Problem and Solution

Cluster: 1
Write a function that receives as argument a month (from 1 to 12) and returns the number of days in the given month (ignore leap years!).

```
def days_in_month(month):
    if month == 2:
        return 28
    elif month in (4,6,9,11):
        return 30
    elif month in (1,3,5,7,8,10, 12):
        return 31
```

39 Problem with solution 29

39.1 TSNE representation

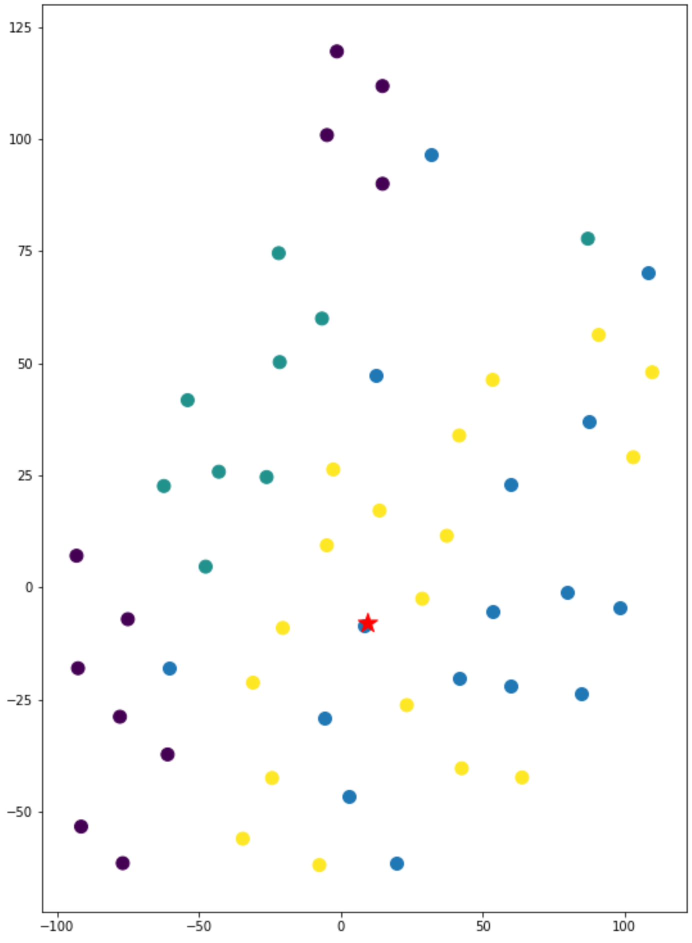

Figure 39: TSNE 2d feature space

39.2 Problem and Solution

Cluster: 2
Create a function that receives as argument a decimal number up to 255 and convert it into its 8-bit binary equivalent.
Your function should return the 8-bit binary as a string, so you can add leading zeros. Make sure your binary number has 8 bits.

```
def dec2bin(decimalNumber):
    binary = ""
    for i in range(7,-1,-1):
        if decimalNumber >= 2**i:
            binary = binary + "1"
            decimalNumber = decimalNumber - 2**i
        else:
            binary = binary + "0"
    return binary
```

40 Problem with solution 36

40.1 TSNE representation

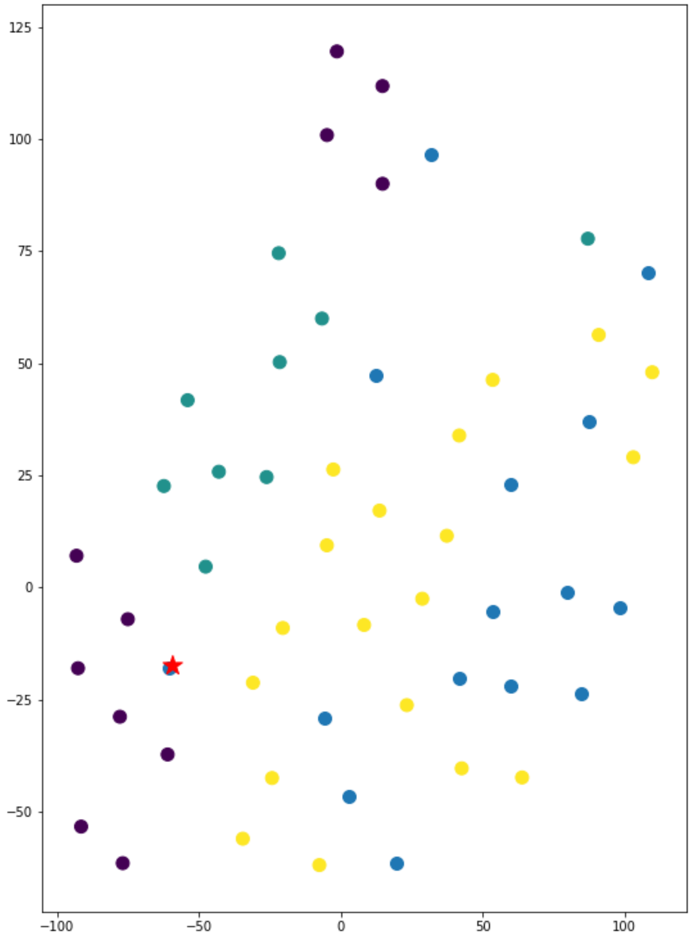

Figure 40: TSNE 2d feature space

40.2 Problem and Solution

Cluster: 0
Write a functions that calculates and returns the value according to the given formula ROUDED OFF to its nearest integer value:
Q = Square root of [(2 * C * D)/H]
Following are the fixed values of C and H:
C is 50. H is 30.
D is the variable whose values should be input to your function as argument.
Example:
Let us assume the following values are given to the function as D:
a) 100
b) 150
c) 180
The output of the function should be, respectively:
a) 18
b) 22
c) 24
Make sure the returned type is integer!

```
def formula(D):
    # So it works using eval
    import math
    C = 50
    H = 30
    Q = round(math.sqrt(2*C*D/float(H)))
    return Q
```

41 Problem with solution 40

41.1 TSNE representation

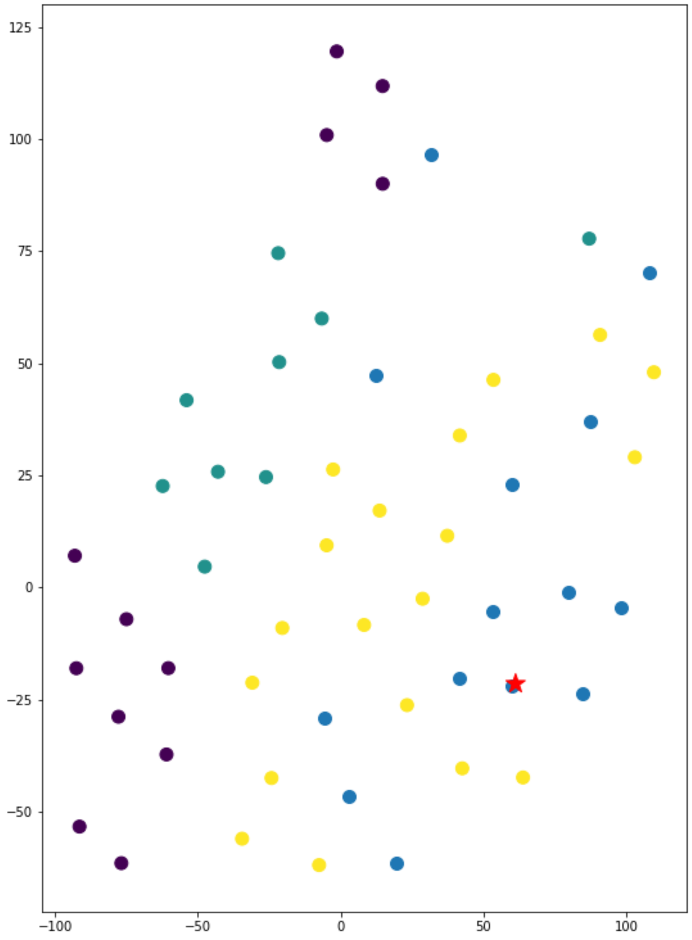

Figure 41: TSNE 2d feature space

41.2 Problem and Solution

Cluster: 2
Write a program that receives a sequence of whitespace separated words as argument and returns the words after removing all duplicate words and sorting them alphanumerically.
Suppose the following input is supplied to the program:
hello world and practice makes perfect and hello world again
Then, the output should be:
again and hello makes perfect practice world

```
def sort_dedupe(words):
    items = words.split(’ ’)
    items_dedupe = []
    for word in items:
        if word not in items_dedupe:
            items_dedupe.append(word)
    items_dedupe.sort()
    return " ".join(items_dedupe)
```

42 Problem with solution 50

42.1 TSNE representation

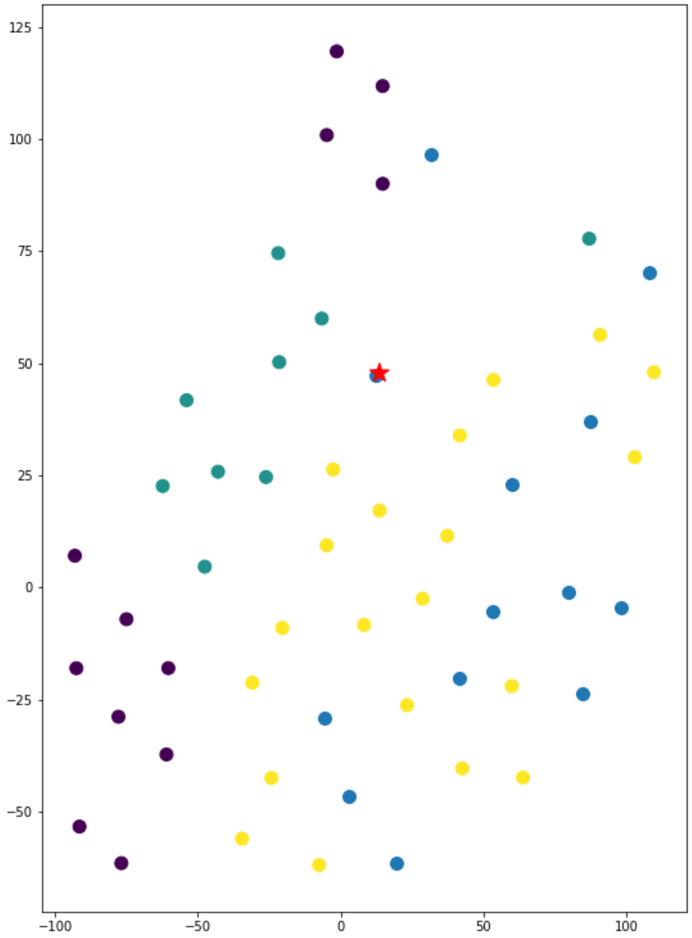

Figure 42: TSNE 2d feature space

42.2 Problem and Solution

Cluster: 1
A website requires the users to input username and password to register.
Write a program to check the validity of password input by users.
Following are the criteria for checking the password:
1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
1. At least 1 letter between [A-Z]
3. At least 1 character from [$#@]
4. Minimum length of transaction password: 6
5. Maximum length of transaction password: 12
Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.
Example
If the following passwords are given as input to the program:
ABd1234@1,a F1#,2w3E*,2We3345
Then, the output of the program should be:
ABd1234@1

```
def check_password(passwords):
    import re
    value = []
    items=[x for x in passwords.split(',')]
    for p in items:
        if len(p)<6 or len(p)>12:
            continue
        else:
            pass
        if not re.search("[a-z]",p):
            continue
        elif not re.search("[0-9]",p):
            continue
        elif not re.search("[A-Z]",p):
            continue
        elif not re.search("[$#@]",p):
            continue
        elif re.search("\s",p):
            continue
        else:
            pass
        value.append(p)
    return ",".join(value)
```

43 Problem with solution 41

43.1 TSNE representation

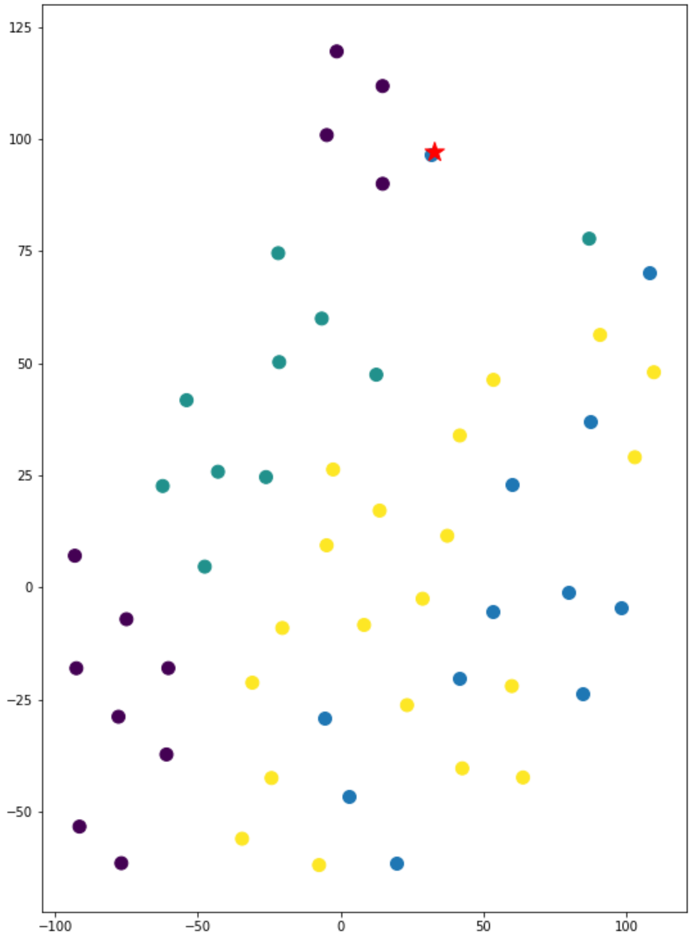

Figure 43: TSNE 2d feature space

43.2 Problem and Solution

Cluster: 0
Write a function which receives a sequence of comma separated 4-digit binary numbers as argument and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be returned in a comma separated sequence.
Example:
0100,0011,1010,1001
Then the output should be:
1010

```
def binby5(number_csv):
    value = []
    items=[x for x in number_csv.split(',')]
    for p in items:
        intp = int(p, 2)
        intp = int(p[0])*8 + int(p[1])*4 + int(p[2])*2 + int(p[3])
        if not intp%5:
            value.append(p)
    return ", ".join(value)
```

44 Problem with solution 27

44.1 TSNE representation

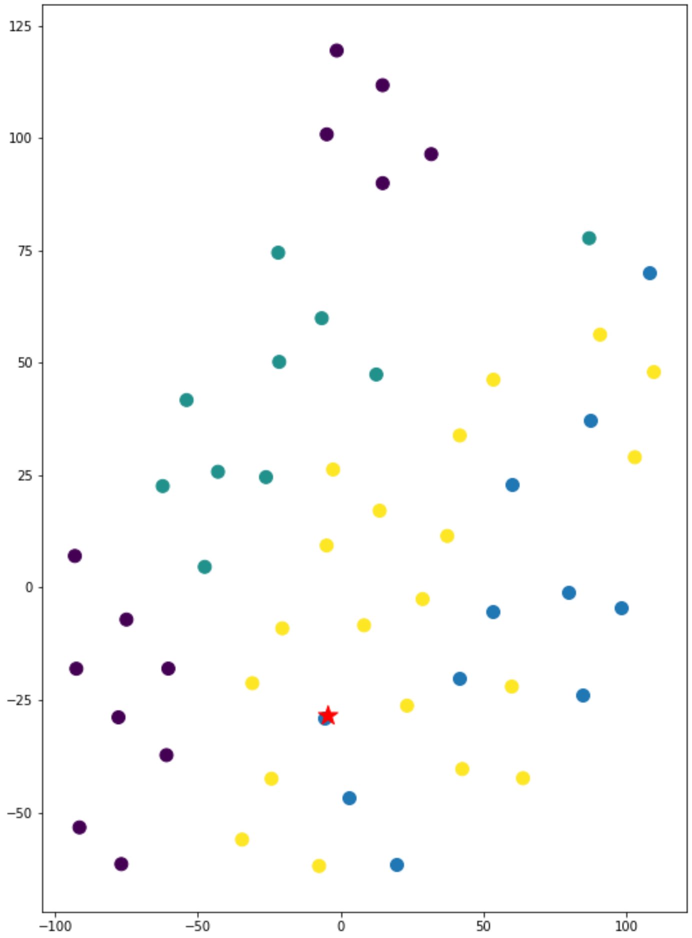

Figure 44: TSNE 2d feature space

44.2 Problem and Solution

Cluster: 2
Create a function that uses linear search to check that a given character is in a given string and replace all characters that are not the searched character with underscore (_). The function receives as argument a string and a character and should return the replaced string if the character was found or return the sentence "Character not found.", otherwise.
1) In a linear search, each character of the given string is compared to the given character. Do not use any built-in operators.
For example, if the string was "hello" and the given character was "l" the program would display "__ll_".

```
def search_and_replace(targetStr, character):
    found = False
    outputStr = ""
    for eachChar in targetStr:
        if eachChar == character:
            outputStr = outputStr + character
            found = True
        else:
            outputStr = outputStr + ’_’

    if not found:
        return "Character not found."
    else:
        return outputStr
```

45 Problem with solution 35

45.1 TSNE representation

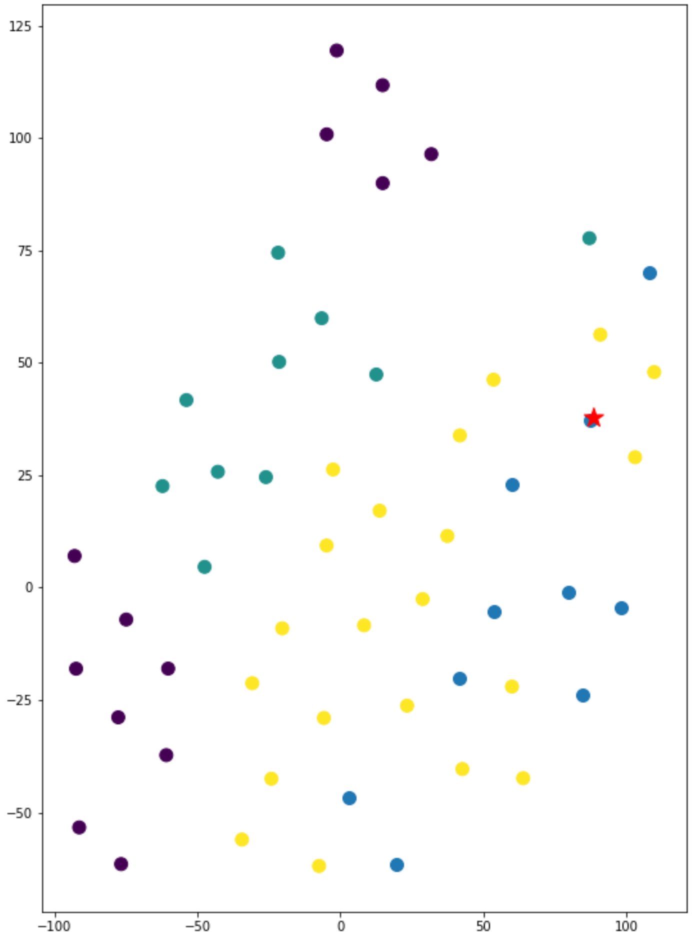

Figure 45: TSNE 2d feature space

45.2 Problem and Solution

Cluster: 2
With a given integral number n, write a function that receives n as an argument and generate a dictionary that contains the number from 1 to n (both included) as keys and their square value as the dictionary value for each respective key. The function should return the dictionary.
Suppose the following input is supplied to the program:
8
Then, the output should be:
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

```
def square(number):
    d=dict()
    for i in range(1,number+1):
        d[i]=i*i
    return d
```

46 Problem with solution 15

46.1 TSNE representation

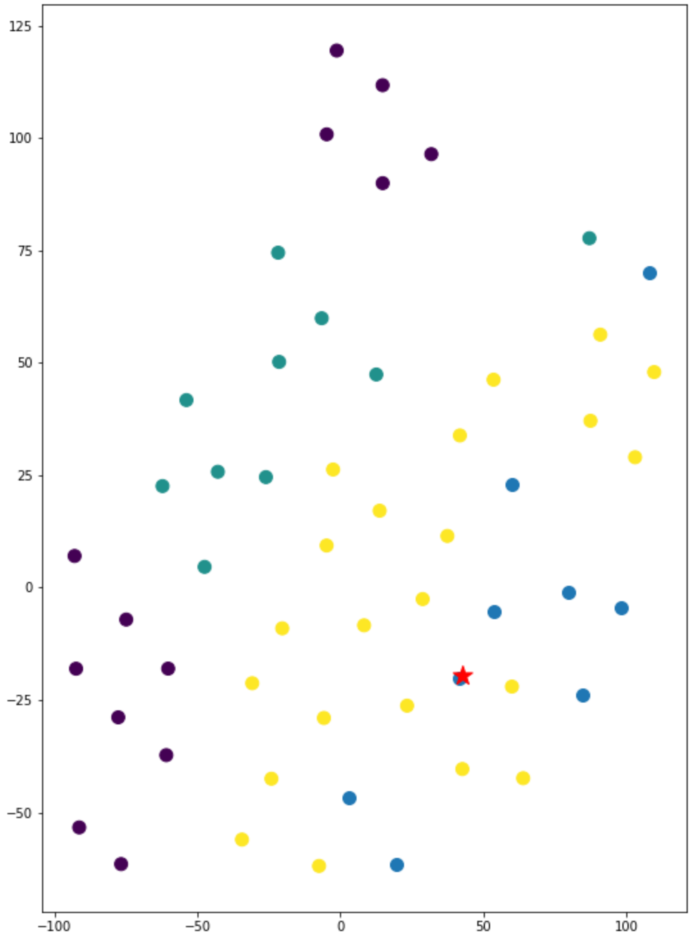

Figure 46: TSNE 2d feature space

46.2 Problem and Solution

Cluster: 2
Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates, keeping the original order.

```
def dedupe(dup_list):
    nodup_list = []
    for i in dup_list:
        if i not in nodup_list:
            nodup_list.append(i)
    return nodup_list
```

47 Problem with solution 26

47.1 TSNE representation

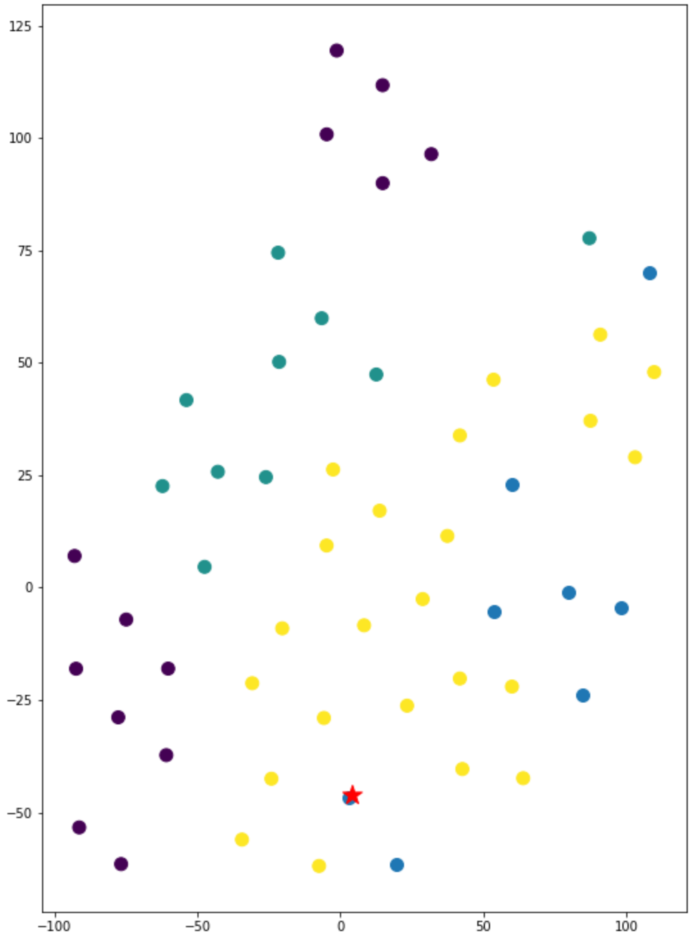

Figure 47: TSNE 2d feature space

47.2 Problem and Solution

Cluster: 2
Create a function that uses linear search to check that a given character is in a given string. The function receives as argument a string and a character and should return the boolean values True or False if the character is found or not in the string, respectivelly. Remember:
1) In a linear search, each character of the given string is compared to the given character. Do not use any built-in operators.

```
def search(targetStr, character):
    found = False
    for eachChar in targetStr:
        if eachChar == character:
            found = True
    if not found:
        return False
    else:
        return True
```

48 Problem with solution 28

48.1 TSNE representation

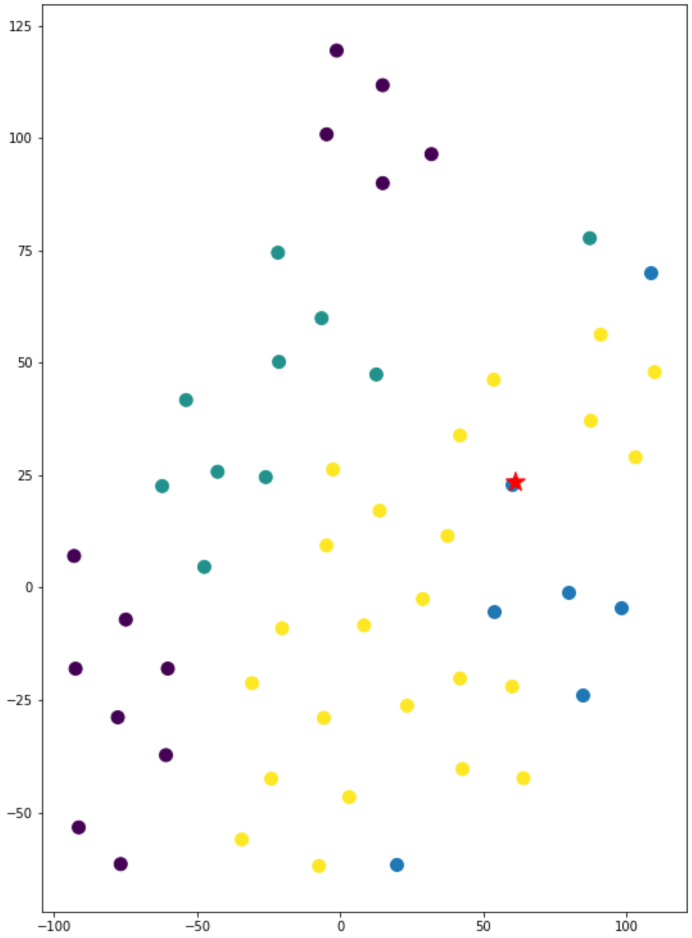

Figure 48: TSNE 2d feature space

48.2 Problem and Solution

Cluster: 2
Create a function that receives as argument a decimal number up to 255 and convert it into its 8-bit binary equivalent.
Your function should return the 8-bit binary as a string, so you can add leading zeros. Make sure your binary number has 8 bits.

```
def dec2bin(decimalNumber):
    binaryString = ""
    if decimalNumber == 0:
        binaryString = "00000000"
    else:
        while decimalNumber > 0:
            #work out whether a 0 or 1 goes in the next position
            positionValue = str(decimalNumber % 2)
            #append to string
            binaryString = positionValue + binaryString
            #
            decimalNumber = decimalNumber // 2

    #make sure number has eight bits
    addZeros = 0
    if len(binaryString) < 8:
        addZeros = 8 - len(binaryString)
    for eachZero in range(addZeros):
        binaryString = "0" + binaryString
    return binaryString
```

49 Problem with solution 13

49.1 TSNE representation

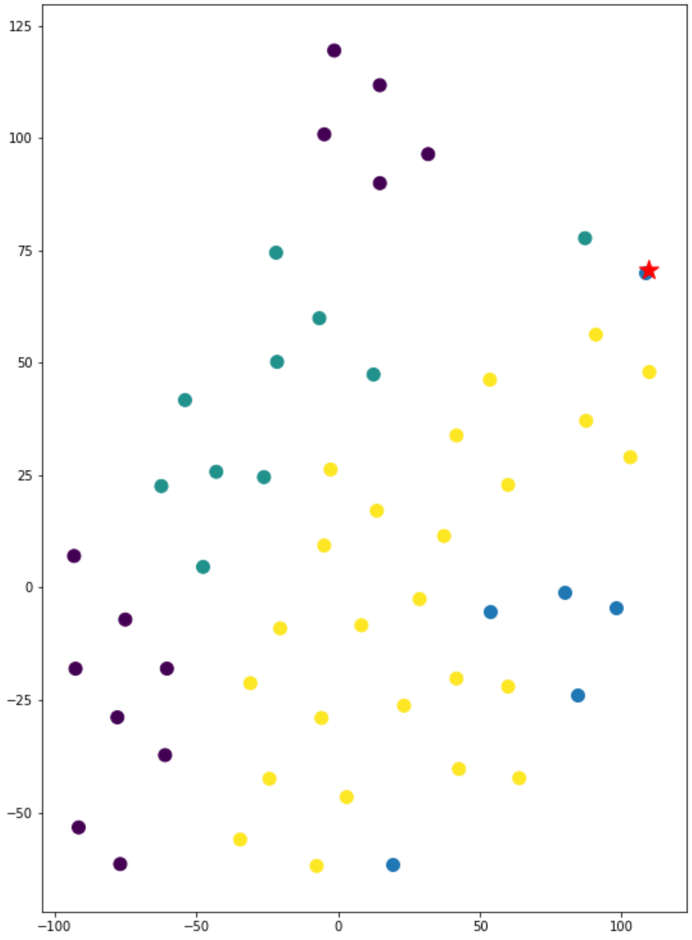

Figure 49: TSNE 2d feature space

49.2 Problem and Solution

Cluster: 2
Given a number as an argument, create a function that determines whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.).

```
def is_prime(number):
    '''Returns True for prime numbers, False otherwise'''
    #Edge Cases
    if number == 1:
        prime = False
    elif number == 2:
        prime = True
    #All other primes
    else:
        prime = True
        for check_number in range(2, int(number/2)+1):
            if number % check_number == 0:
                prime = False
                break
    return prime
```

50 Problem with solution 9

50.1 TSNE representation

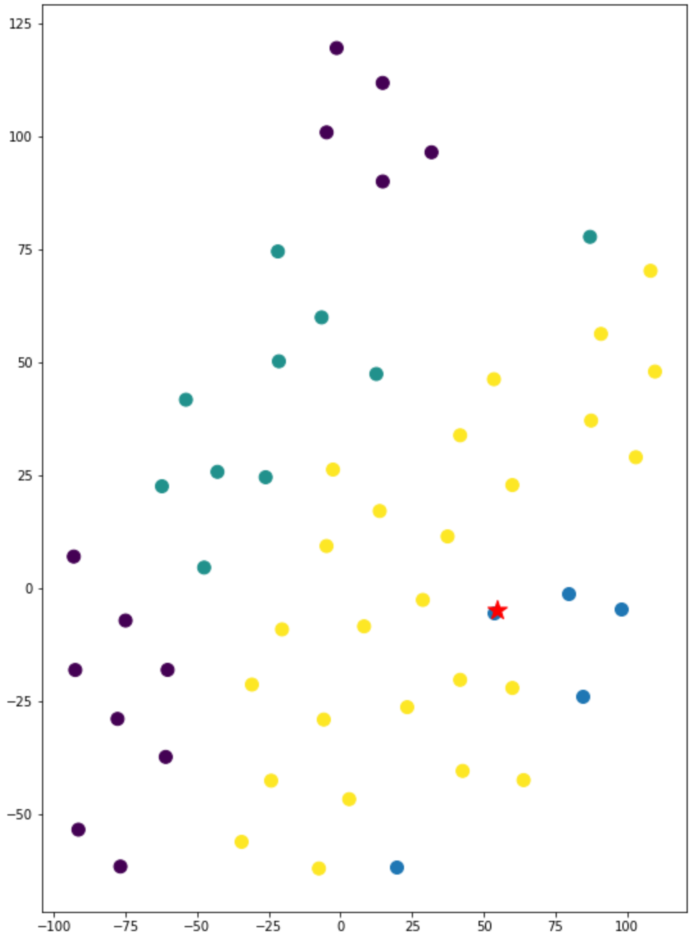

Figure 50: TSNE 2d feature space

50.2 Problem and Solution

Cluster: 2
Take two lists, say for example these two:
list1 = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
list2 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
and write a function that returns a list that contains only the elements that are common between the lists (without duplicates and ordered ascendantly). Make sure your program works on two lists of different sizes.

```
def common(list1, list2):
    common_list = []
    for i in list1:
        if i in list2 and i not in common_list:
            common_list.append(i)

    common_list.sort()
    return common_list
```