Homework 3

ECE204 Data Science & Engineering

Question 1

> A number N is *divisible* by P if P divides N evenly. For example, 15 is divisible by 1, 3, 5, and 15. The numbers [1,3,5,15] are called the "divisors" of 15. Your task is to create a function that returns the sum of all the divisors of a number. For example, if you call the function "F", you should find that F(15) returns 24. We can figure out if N is divisible by P by using Python's "modulo" function. The syntax for this is:
>
> ```
> N % P == 0    # this returns True if P divides N
> ```
>
> For more information about the modulo (%) function in Python, see this link (https://www.pythoncentral.io/using-python-to-check-for-odd-or-even-numbers/).
>
> Write the function "F" and test it to make sure that `F(15) == 24`. Call the function with the argument given to you in Canvas.

In [5]:

```
# your code here
def F(n):
    sum = 0
    for i in range(1, int(n / 2) + 1):
        if n % i == 0:
            sum += i
    sum += n
    return sum

F(13860)
```

Out[5]:

```
52416
```

Question 2

List comprehensions we saw in class looked like this:

```
[ (do this) for (item in collection) ]
```

But what if we only want to select the items that satisfy an additional condition as well? We can use the syntax:

```
[ (do this) for (item in collection) if (true/false condition) ]
```

Use this technique to find the sum of numbers divisible by 7 or 11 and report this result. Be sure to use the range of vlaues (inclusive) given to you in Canvas. You may want to start with a smaller number first to make sure your list comprehension is doing the right thing!

```
In [11]:
# your code here
sum([ i for i in range (61, 10026) if ((i%7==0) or (i%11==0)) ])
```

```
Out[11]: 11095700
```

Question 3

The square of the sum of the first ten natural numbers is

$$(1+2+\ldots+10)^2 = 55^2 = 3025$$

The sum of the squares of the first ten natural numbers is

$$1^2 + 2^2 + \ldots + 10^2 = 385$$

Hence the difference between the square of the sum of first ten natural numbers and the sum of the squares is 3025 − 385 = 2640.

Find the difference between the square of the sum of the first X natural numbers and the sum of the squares. Note that X is the value given to you in Canvas

In [17]:

```python
# your code here
def F(n):
    square_of_sum = sum([i for i in range(n+1)])**2
    sum_of_square = sum([i**2 for i in range(n+1)])
    difference = square_of_sum - sum_of_square
    return difference

F(120)

```

Out[17]:

```
52124380
```

Question 4

> Out of all the points in the list of `candidates`, find the one that is the furthest (in Euclidean distance) from the `given` point? Note: the `given` point is provided for you in Canvas. What is this furthest / highest distance? **Report this furthest distance rounded to two numbers after the decimal.**
>
> `Hint: np.linalg.norm( )`

In [19]:

```python
candidates = [
    (0.0, 1.0),
    (0.32, 0.95),
    (0.61, 0.79),
    (0.84, 0.55),
    (0.97, 0.25),
    (1.0, -0.08),
    (0.92, -0.4),
    (0.74, -0.68),
    (0.48, -0.88),
    (0.16, -0.99),
    (-0.16, -0.99),
    (-0.48, -0.88),
    (-0.74, -0.68),
    (-0.92, -0.4),
    (-1.0, -0.08),
    (-0.97, 0.25),
    (-0.84, 0.55),
    (-0.61, 0.79),
    (-0.32, 0.95),
    (-0.0, 1.0)
]

given = [3, 1]
```

In [26]:

```python
# your code here
import numpy as np
sorted([ [np.linalg.norm(np.array(given) - np.array(i)), i] for i in
```

Out[26]:

```
[[2.164116447883524, (0.97, 0.25)],
 [2.2063771209836274, (0.84, 0.55)],
 [2.2729716232280595, (1.0, -0.08)],
 [2.3992082027202226, (0.61, 0.79)],
 [2.5072694310743713, (0.92, -0.4)],
 [2.680466377330632, (0.32, 0.95)],
 [2.8160255680657444, (0.74, -0.68)],
 [3.0, (0.0, 1.0)],
 [3.0, (-0.0, 1.0)],
 [3.1440101781005736, (0.48, -0.88)],
 [3.3203764846776034, (-0.32, 0.95)],
 [3.467809106626257, (0.16, -0.99)],
 [3.6161028746428108, (-0.61, 0.79)],
 [3.7343941945113404, (-0.16, -0.99)],
 [3.8662772792442084, (-0.84, 0.55)],
 [3.9553508061864755, (-0.48, -0.88)],
 [4.0402227661355505, (-0.97, 0.25)],
 [4.1000000000000005, (-0.74, -0.68)],
 [4.143235450707575, (-1.0, -0.08)],
 [4.162499249249182, (-0.92, -0.4)]]
```

Question 5

> `celsius.csv` contains a list of temperatures in Celsius. **What is the mean Fahrenheit temperature? (Report your answer rounded to two numbers after the decimal)**
>
> The Celsuis to Fahrenheit conversion is given by `fahrenheit = 1.8 * celsius + 32`.

In [28]:

```python
# given code
import pandas as pd
# Import the data file and have a look!
s = pd.read_csv("celsius.csv")
s.head()
```

Out[28]:

	temps
0	54.88
1	71.52
2	60.28
3	54.49
4	42.37

Question 6

> **What is the fastest method to convert from Celsius to Fahrenheit?**
>
> - a list comphrension used to make a list of all the converted temperatures
> - using `Series.apply` to convert each element of the Series
> - directly evaluating (`1.8*T + 32`), where `T` is the Series of temperatures
>
> `NOTE:` Use `%timeit` to do your timing analysis. Use the `celsius.csv` data file from the previous problem.

In [47]:

```python
# your code here

# Define the conversion functions
def method1(T):
    return 1.8 * T + 32

def method2(T):
    return T.apply(lambda x: 1.8 * x + 32)

def method3(T):
    return [1.8 * x + 32 for x in T]

# Use %timeit to measure the execution time for each method
%timeit -n 100 method1(s['temps'])
%timeit -n 100 method2(s['temps'])
%timeit -n 100 method3(s['temps'])

```

```
71.3 µs ± 11.3 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
992 µs ± 10.9 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
705 µs ± 12.4 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

Question 7

What is one of the centers in `features.csv` when using Scikit-Learn's KMeans? They are listed as `[feature0, feature1]`, and `KMeans` should be used as `KMeans(n_clusters=4, random_state=42)`. These numbers are rounded -- pick the closest one in the list.

Hint: read the documentation at https://scikit-

In [135]:

```
# Read the dataset and take a look!
df = pd.read_csv("features.csv")
df.head()
```

Out[135]:

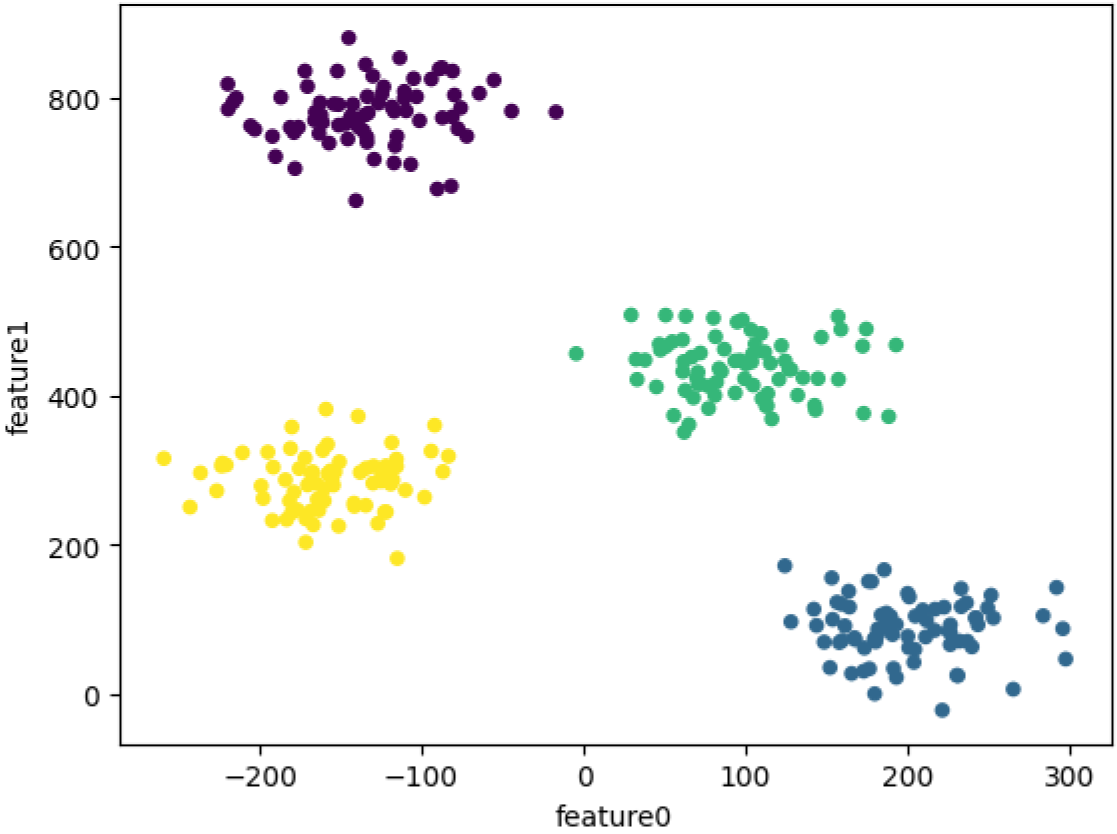
	feature0	feature1
0	124.30	172.35
1	-135.85	755.16
2	109.56	483.43
3	-109.52	782.18
4	153.27	156.09

In [140]:

```python
# your code here

from sklearn.cluster import KMeans
km = KMeans(n_clusters=4, random_state=42)

km.fit(df)

# Get the cluster centers
centers = km.cluster_centers_


df.plot.scatter(x="feature0", y="feature1", c=km.predict(df),cmap="v
print(centers)
```

```
[[-133.15813333  778.09466667]
 [ 200.6812       87.76973333]
 [  95.85786667  438.06426667]
 [-156.52253333  285.9836    ]]
```

```
/Users/janeli/anaconda3/envs/bonding/lib/python3.11/site-packages/sklea
rn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explici
tly to suppress the warning
  warnings.warn(
```



Question 8

> **Using the same setup as before, which cluster center is the point `[-100, 850]` nearest to?**
>
> `NOTE:` **choose a valid cluster center**
>
> - [-157, 286]
> - [-91, 743]
> - [201, 87]

In [52]:

```python
# your code here
point = [-100, 850]
arrays = [[ -133, 778],[-157, 286],[-91, 743],[ 201, 87]]
sorted([ [np.linalg.norm(np.array(point) - np.array(i)), i] for i in
```

Out[52]:

```
[[79.20227269466452, [-133, 778]],
 [107.37783756436893, [-91, 743]],
 [566.8730016502815, [-157, 286]],
 [820.2255787281936, [201, 87]]]
```

In [65]:

```python
import random
r = lambda x: random.randint(0, 99)
random.seed(0)

names_and_ages = {'Alice': 25, 'Bob': 18, 'Charlie': 32, 'David': 9,
# Add more names and ages to the dictionary
names_and_ages.update({'Frank': r(0), 'Grace': r(0), 'Henry': r(0),
result = []
for key in names_and_ages:
    age = names_and_ages[key]
    if (age >= 20 and age <= 29):
        result.append(key)
result

```

Out[65]:

```
['Alice', 'Ryan', 'Tara']
```

In [68]:

```python
result = []
for key in names_and_ages:
    age = names_and_ages[key]
    if (age >= 30 and age <= 39):
        result.append(key)
result
```

Out[68]:

```
['Noah', 'Uma', 'Eli', 'Jade', 'Paige']
```

In [69]:

```
result = []
for key in names_and_ages:
    age = names_and_ages[key]
    if (age >= 35 and age <= 44):
        result.append(key)
result
```

Out[69]:

```
['Noah', 'Uma', 'Eli', 'Jade', 'Paige']
```

In [71]:

```
string = ('Data science is the art and science of extracting insight
          'numbers, texts, images, sounds, or even DNA. Data science
          'learning, programming, visualization, and domain knowledg
          'can be applied to many fields and problems, such as busin
          'science can help us understand the past, present, and fut
          'science can also help us make better decisions, predictio
          'science applications are: Recommending products, movies, 
          'fraud, spam, or anomalies in transactions, emails, or net
          'faces, emotions, or genres. Generating new content, such 
          'solutions, such as routes, schedules, or prices, for comp
          'field, as it can wow us with its power and potential. Dat
          'many people and sectors. Data science is not a racecar or
```

In [91]:

```
sentences = string.split(". ")
words = []
for sentence in sentences:
    sentence = sentence.split(", ")
    for s in sentence:
        words.append(s)

all_word = []
for word in words:
    word = word.split(" ")
    for w in word:
        all_word.append(w)

palindrome = []
for word in all_word:
    word = word.lower()
    if (word == word[::-1]):
        palindrome.append(word)

palindrome
```

Out[91]:

```
['a', 'civic', 'radar', 'wow', 'a', 'a', 'racecar', 'a', 'rotor', 'a',
'kayak']
```

In [97]:

```
string = ('Data science is the discipline of discovering, deriving, 
          'Data science is not only about analyzing data, but also al
```

In [98]:

```python
sentences = string.split(". ")
words = []
for sentence in sentences:
    sentence = sentence.split(", ")
    for s in sentence:
        words.append(s)

all_word = []
for word in words:
    word = word.split(" ")
    for w in word:
        all_word.append(w)

special = []
for word in all_word:
    if (word[0] == "d" and word[1] != "a"):
        special.append(word)

len(special)
```

Out[98]:

```
14
```

In [99]:

```python
s = ('Data science is the field of study that combines mathematics, 
```

In [118]:

```python
s = s.replace("data", "D")
s[2294:2400]
```

Out[118]:

```
'science is a vital and essential field of study that can help us make
sense of the D that surrounds us, cr'
```

In [133]:

```
import re

def count_valid_us_phone_numbers(phone_numbers):
    # Regular expression pattern for valid US phone numbers
    pattern = r'\(\d{3}\) \d{3}-\d{4}'
    # Count of valid US phone numbers
    valid_count = 0
    # Iterate through each phone number
    for phone_number in phone_numbers:
        # Check if the phone number matches the pattern and has exac
        if re.fullmatch(pattern, phone_number):
            # Increment the count if it's valid
            valid_count += 1
    return valid_count





phone_numbers = ['(7425) 734-9527', '(299) 937-6414', '(994) 285-738

count_valid_us_phone_numbers(phone_numbers)


```

Out[133]:

```
204
```

In [134]:

```
passwords = ['M9TB!My6@', 'Q8mT3Zi8?', 'Za$QFPo8!', '@nMLpQ0Hh2&', '
```

In [174]:

```
def count_valid_passwords(passwords):
    # Regular expression patterns for password requirements
    patterns = [
        r'.{8,}',           # At least 8 characters long
        r'.*[A-Z].*',       # At least one uppercase letter
        r'.*[a-z].*',       # At least one lowercase letter
        r'.*\d.*',          # At least one digit
        r'.*[@$!%*?&].*'   # At least one special character
    ]
    # Compile patterns into regex objects
    compiled_patterns = [re.compile(pattern) for pattern in patterns
    # Count of valid passwords
    valid = []
    # Iterate through each password
    for password in passwords:
        # Check if the password matches all patterns
        if all(pattern.match(password) for pattern in compiled_patte
            # Increment the count if it's valid
            valid.append(password)
    return valid

len(count_valid_passwords(passwords))
```

Out[174]:

```
589
```

In [166]:

```
primes = []

# Function to generate N prime numbers using
# Sieve of Eratosthenes
def SieveOfEratosthenes():

    n = 1000005

    # Create a boolean array "prime[0..n]" and
    # initialize all entries it as true. A value
    # in prime[i] will finally be false if i is
    # Not a prime, else true.
    prime = [True for i in range(n + 1)]

    p = 2
    while (p * p <= n):

        # If prime[p] is not changed,
        # then it is a prime
        if (prime[p] == True):

            # Update all multiples of p
            for i in range(p * p, n + 1, p):
                prime[i] = False

        p += 1

    # Print all prime numbers
    for p in range(2, n + 1):
        if prime[p]:
            primes.append(p)

def isPowerofTwo(n):

    if (n == 0):
        return 0
    if ((n & (~(n - 1))) == n):
        return 1
    return 0


# Driver code
if __name__=='__main__':

    # Function call
    SieveOfEratosthenes()
    sum = 0
    for i in range (50000):
    #    print("prime:" + str(primes[i]))
       if isPowerofTwo(primes[i]+1):
           print(primes[i])


```

```
3
7
31
127
8191
131071
524287
```

In [152]:

```python
def nth_prime(n):
    """
    This function finds the nth prime number.
    """
    count = 0
    num = 2
    while count < n:
        if is_prime(num):
            count += 1
        if count == n:
            return num
        num += 1
def is_prime(num):
    """
    This function checks if a number is prime or not.
    """
    if num <= 1:
        return False
    elif num <= 3:
        return True
    elif num % 2 == 0 or num % 3 == 0:
        return False
    i = 5
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            return False
        i += 6
    return True




sum = 0
for i in range (11,190):
    sum += nth_prime(i)**2

print(sum)
```

```
73280915
```

In [170]:

```
import math

def find_n(number):
    n = 1
    while True:
        result = int(math.sqrt(n**2 * 2**n)) + 1
        if result == number:
            return n
        elif result > number:
            return -1
        n += 1

# Test the function
number = 2097153
result = find_n(number)
print("Result:", result)


```

```
Result: 32
```

In [ ]:

```
def sum_odd_lengths_in_latex(text):
    import re

    # Remove comments
    no_comments = re.sub(r"%.*", "", text)

    # Find all pairs of parentheses and brackets, considering the ru
    # Using non-greedy matching to handle nested structures correctl
    pairs = re.findall(r"\(([^()]*)\)|\{([^\{\}]*)\}", no_comments)

    # Flatten the list of tuples and filter out empty strings
    contents = [content for pair in pairs for content in pair if con

    # Calculate the number of characters inside each pair
    lengths = [len(content) for content in contents]

    # Sum the elements that are odd numbers
    sum_odds = sum(length for length in lengths if length % 2 == 1)

    return sum_odds

latex_code = r"""
\documentclass{article}
\usepackage{amsmath, amssymb, graphicx}

\begin{document}
\title{Testing out Latex}
\author{204}
\date{\today}
\maketitle

\section*{Introduction to Machine Learning and Transformers}

Machine learning (ML) is a transformative field within artificial in

One powerful paradigm in ML is the transformer architecture, which ha

\subsection*{Attention Mechanism}

The attention mechanism is a key component of transformers. Given a 

The attention mechanism can be defined as:
\begin{equation}
    \text{{Attention}}(Q, K, V) = \text{{softmax}}\left(\frac{QK^T}{
\end{equation}
Here, $Q$, $K$, and $V$ are matrices representing the query, key, and

\subsection*{Transformer Model}
The transformer model is composed of an encoder and a decoder, each 

The output of the self-attention mechanism in a transformer layer is
\begin{equation}
    \text{{Output}} = \text{{LayerNorm}}(\text{{Input}} + \text{{Att
\end{equation}
Here, $\text{{LayerNorm}}$ is layer normalization, and $\text{{Input}

\subsection*{Applications}
```

```

Transformers have achieved remarkable success in various application

In conclusion, machine learning, with the transformative power of mo

% % \emph{Important} remark: % \textbf{ECE } \emph{for 204}
\textbf{Remark}: We emphasize that this is a sample text and it shou

\textbf{Nevertheless, this is an exciting era % for all of ML
and we do recommend that you consider this as a starting point to di

\emph{\LARGE{Copyright: (204) 2024 (ECE)}}
\textbf{20 %
24}
\bibliographystyle{plain}
\bibliography{references}

\end{document}
"""
sum_odd_lengths_in_latex(latex_code)
```

In [181]:

```
import re

text = """Data science and machine learning are two interrelated fie

One of the key factors that drives the success and innovation of dat

However, not all data is created equal. The quantity, diversity, and

One of the most prominent examples of how data can transform data sc

ImageNet has been instrumental in advancing the field of computer vi

One of the most influential events in the history of computer vision

The ILSVRC witnessed a dramatic improvement in the performance and a

AlexNet sparked a revolution in computer vision and deep learning, a

ImageNet and the ILSVRC have demonstrated the power and potential of

- Data quality and diversity: ImageNet and the ILSVRC are based on a

- Data availability and accessibility: ImageNet and the ILSVRC are p

- Data interpretation and explanation: ImageNet and the ILSVRC are f

Data science and machine learning are two exciting and promising fie

# Define the regular expression pattern
pattern = r'\bdata\b(?! science)'

# Compile the pattern with the re.IGNORECASE flag to make the search
regex = re.compile(pattern, flags=re.IGNORECASE)

# Find all matches in the text
matches = regex.findall(text)

# Print the matches
len(matches)

```

Out[181]:

```
28
```