DATA WAREHOUSING AND DATA MINING LAB (CSD-421) LAB ASSIGNMENT 1 & 2

AKSHAT RAJ VANSH (185520)

January 27, 2022

Contents

1 Program 1

1.1 Question

L is a list defined as L= [11, 12, 13, 14].
i. WAP to add 50 and 60 to L.
ii. WAP to remove 11 and 13from L.
iii. WAP to sort L in ascending order.
iv. WAP to sort L in descending order.
v. WAP to search for 13 in L.
vi. WAP to count the number of elements present in L.
vii. WAP to sum all the elements in L.
viii. WAP to sum all ODD numbers in L.
ix. WAP to sum all EVEN numbers in L.
x. WAP to sum all PRIME numbers in L.
xi. WAP to clear all the elements in L.
xii. WAP to delete L.

1.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


def isPrime(x):
    if x > 1:
        for i in range(2, x):
            if x % i == 0:
                return False
        return True
    return False


def main():
    L = [11, 12, 13, 14]

    L.extend([50, 60])
    print('i.   L: {}'.format(L))

    L = [x for x in L if [11, 13].count(x) == 0]
    print('ii.  L: {}'.format(L))

    L.sort(reverse=False)
    print('iii. L: {}'.format(L))

    L.sort(reverse=True)
    print('iv.  L: {}'.format(L))

    pos = L.count(13) if L.count(13) != 0 else 'Not Present'
    print('v.   Position of 13: {}'.format(pos))

    print('vi.  Length of L: {}'.format(len(L)))

    print('vii. Sum of all elements: {}'.format(sum(L)))

```

```python
    odd = [x for x in L if x % 2 != 0]
    print('viii.Sum of all ODD numbers in L: {}'.format(sum(odd)))

    even = [x for x in L if x % 2 == 0]
    print('ix.  Sum of all EVEN numbers in L: {}'.format(sum(even)))

    print('x.   Sum of all PRIME numbers in L: {}'.format(sum(x for x in L if isPrime(x))))

    print('xi.  Clearing all elements in L: {}'.format(L.clear()))

    del L


if __name__ == '__main__':
    main()
```

1.3 Output

Question 1 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 1> python .\main.py
i.   L: [11, 12, 13, 14, 50, 60]
ii.  L: [12, 14, 50, 60]
ii.  L: [12, 14, 50, 60]
iii. L: [12, 14, 50, 60]
iv.  L: [60, 50, 14, 12]
v.   Position of 13: Not Present
vi.  Length of L: 4
vii. Sum of all elements: 136
viii.Sum of all ODD numbers in L: 0
ix.  Sum of all EVEN numbers in L: 136
x.   Sum of all PRIME numbers in L: 0
xi.  Clearing all elements in L: None
```

2 Program 2

2.1 Question

D is a dictionary defined as D= 1:5.6, 2:7.8, 3:6.6, 4:8.7, 5:7.7.
i. WAP to add new entry in D; key=8 and value is 8.8
ii. WAP to remove key=2.
iii. WAP to check weather 6 key is present in D.
iv. WAP to count the number of elements present in D.
v. WAP to add all the values present D.
vi. WAP to update the value of 3 to 7.1.
vii. WAP to clear the dictionary.

2.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


def main():
    D= {1:5.6, 2:7.8, 3:6.6, 4:8.7, 5:7.7}
    D[8] = 8.8
    print('i.   D: {}'.format(D))
    D.pop(2)
    print('ii.  D: {}'.format(D))
    pos = [x for x in D.keys()].count(6)
    print('iii. Present: {}'.format(pos if pos>0 else 'Not Present'))
    print('iv.  Number of elements in D: {}'.format(len(D.values())))
    sumofDict = sum([x for x in D.values()])
    print('v.   Sum of all values: {}'.format(sumofDict))
    D[3] = 7.1
    print('vi.  D: {}'.format(D))
    print('vii. Clearing all elements in L: {}'.format(D.clear()))


if __name__ == '__main__':
    main()
```

2.3 Output

Question 2 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 2\Codes> python .\main.py
i.   D: {1: 5.6, 2: 7.8, 3: 6.6, 4: 8.7, 5: 7.7, 8: 8.8}
ii.  D: {1: 5.6, 3: 6.6, 4: 8.7, 5: 7.7, 8: 8.8}
iii. Present: Not Present
iv.  Number of elements in D: 5
v.   Sum of all values: 37.4
vi.  D: {1: 5.6, 3: 7.1, 4: 8.7, 5: 7.7, 8: 8.8}
vii. Clearing all elements in L: None
```

3 Program 3

3.1 Question

S1 is a set defined as S1= [10, 20, 30, 40, 50, 60].
S2 is a set defined as S2= [40, 50, 60, 70, 80, 90].
i. WAP to add 55 and 66 in Set S1.
ii. WAP to remove 10 and 30 from Set S1.
iii. WAP to check whether 40 is present in S1.
iv. WAP to find the union between S1 and S2.
v. WAP to find the intersection between S1 and S2.
vi. WAP to find the S1 - S2.

3.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


def main():
    S1= {10, 20, 30, 40, 50, 60}
    S2= {40, 50, 60, 70, 80, 90}

    S1.update({55,66})
    print('i.   S1: {}'.format(S1))
    S1.difference_update([10,30])
    print('ii.  S1: {}'.format(S1))
    pos = [x for x in S1].count(40)
    print('iii. Present: {}'.format("Present" if pos>0 else 'Not Present'))
    print('iv.  Union of S1 and S2: {}'.format(S1.union(S2)))
    print('v.   Intersection of S1 and S2: {}'.format(S1.intersection(S2)))
    print('vi.  S1-S2: {}'.format(S1-S2))

if __name__ == '__main__':
    main()
```

3.3 Output

Question 3 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 3\Codes> python .\main.py
i.   S1: {50, 66, 20, 55, 40, 10, 60, 30}
ii.  S1: {50, 66, 20, 55, 40, 60}
iii. Present: Present
iv.  Union of S1 and S2: {66, 70, 40, 80, 50, 20, 55, 90, 60}
v.   Intersection of S1 and S2: {40, 50, 60}
vi.  S1-S2: {66, 20, 55}
```

4 Program 4

4.1 Question

Write the following program.
i. WAP to print 100 random strings whose length between 6 and 8.
ii. WAP to print all prime numbers between 600 and 800.
iii. WAP to print all numbers between 100 and 1000 that are divisible by 7 and 9.

4.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


import random
import string


def randomString():
    print('i. 100 random strings whose length is between 6 to 8: ')
    for i in range(1, 101):
        str = ''.join(random.choices(string.ascii_letters +
                      string.digits, k=random.choice([6, 7, 8])))
        print(str, end=' ')
        if i % 10 == 0:
            print('')


def isPrime(x):
    if x > 1:
        for i in range(2, x):
            if x % i == 0:
                return False
        return True
    return False


def divisibleBy7and9(x):
    if x % 7 == 0 and x % 9 == 0:
        return True
    return False


def printPrime():
    print('ii. All Prime numbers between 600 to 800: ')
    for i in range(600, 801):
        if isPrime(i):
            print(i, end=' ')


def printNumbers():
    print('iii. All Numbers between 100 and 1000 that are divisible by 7 and 9: ')
    for i in range(100, 1001):
        if divisibleBy7and9(i):
            print(i, end=' ')
```

```python


def main():
    #    randomString()
    #    printPrime()
    printNumbers()


if __name__ == '__main__':
    main()
```

4.3 Output

Question 4 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 4\Codes> python .\main.py
i. 100 random strings whose length is between 6 to 8:
kwE2zxo CfOHHN7 uhUsEi 2WRwGDv1 FiDSqU RATCkr K0BGI1 gyb4J1K H1YD0j 4ZzM9C
aTItgT c8wvPA fRehN4B HNEzIX8 5392gQl ubAUJ3m 14tRKCg fvTFm3P 9MVqGPN Z3vZCS
RDzraJMR 3t8XWDsH GakVKhQV 9Tt2TJz KBWuc8Z RPAsk5LV ipvBkz8 6UTwNYB SPaAXn H8SIchA
pM3g3WW 00EmnhO VlbY7Dw UxVKyLQ Z1ZZzJN eRNgupy pXORmGem SMlhKIm KW2eTua ifsoHcRm
ZqCNHn 4m9a5w8w H5YhmH SJ1w5i 4sSXyx 4MnLrWwj 1kOj1m0 RLIJvN OiaL4Q jVLIsjvx
bXnywZR BfZ3qZV PFpcJ3J1 GpaGLLL twkNb3D sK1p9s FKai5b PM0GwZP LBVBtMYw 5URfENj
uUSGvG 6EBgeR MsOogrhZ 2CuxeKGx 4Atq3F8 ocodzuA spmaP0Ej aUdXnc2 ByRDxs5 17h1ruz8
1bLVrI1 vYAUsH RX0xpJ pJh8zZG 4IFAlez MxOT3dh XW5yCv MkhtaryO Nr6IsaK4 04qJ3mZ
IJVWns PBNtri JwCVQ7 L1nMXBK ReVo2y XGP85lk lEdIrPxK PY8P8QWN T9PZkdw 6DoGF4Zy
3BznNWr mvFrehxb WO0NyaQ KyPVfu 8Iu9wXo2 RfsxVI QTOMk6 YnoltR u7shhQ MfhuXGQ
```

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 4\Codes> python .\main.py
ii. All Prime numbers between 600 to 800:
601 607 613 617 619 631 641 643 647 653 659 661 673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797
```

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 4\Codes> python .\main.py
iii. All Numbers between 100 and 1000 that are divisible by 7 and 9:
126 189 252 315 378 441 504 567 630 693 756 819 882 945
```

5 Program 5

5.1 Question

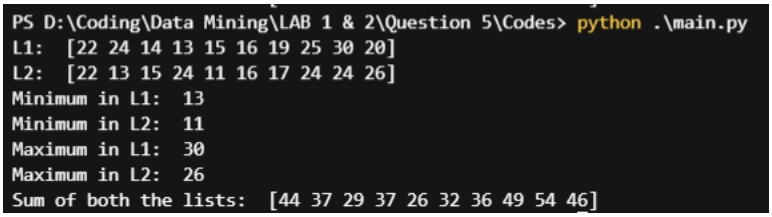
WAP to create two lists of 10 random numbers between 10 and 30; Find
i. Common numbers in the two lists
ii. Unique numbers in both the list
iii. Minimum in both the list
iv. Maximum in both the list
v. Sum of both the lists

5.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'

import numpy as np

def main():
    L1 = np.random.randint(10, 31, 10)
    L2 = np.random.randint(10, 31, 10)
    print('L1: ',L1)
    print('L2: ',L2)
    print('Minimum in L1: ', np.min(L1))
    print('Minimum in L2: ', np.min(L2))
    print('Maximum in L1: ', np.max(L1))
    print('Maximum in L2: ', np.max(L2))
    print('Sum of both the lists: ', L1+L2)


if __name__ == '__main__':
    main()
```

5.3 Output

Question 5 Outputs



```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 5\Codes> python .\main.py
L1:  [22 24 14 13 15 16 19 25 30 20]
L2:  [22 13 15 24 11 16 17 24 24 26]
Minimum in L1:  13
Minimum in L2:  11
Maximum in L1:  30
Maximum in L2:  26
Sum of both the lists:  [44 37 29 37 26 32 36 49 54 46]
```

6 Program 6

6.1 Question

WAP to create a list of 100 random numbers between 100 and 900. Count and print the:
i. All odd numbers
ii. All even numbers
iii. All prime numbers

6.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'

import numpy as np

def isPrime(x):
    if x > 1:
        for i in range(2, x):
            if x%i==0: return False
        return True
    return False


def main():
    L = np.random.randint(100, 901, 100)
    print('L: ',L)

    print('i.  All odd numbers: ')
    _odd = list(filter(lambda x: x%2!=0, L))
    _count = len(_odd)
    print('     Count: ', _count)
    print('     List: ', _odd)

    print('ii. All odd numbers: ')
    _even = list(filter(lambda x: x%2==0, L))
    _count = len(_even)
    print('     Count: ', _count)
    print('     List: ', _even)

    print('iii. All prime numbers: ')
    _primes = list(filter(lambda x: isPrime(x), L))
    _count = len(_primes)
    print('     Count: ', _count)
    print('     List: ', _primes)


if __name__ == '__main__':
    main()
```

6.3 Output

Question 6 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 6\Codes> python .\main.py
L:  [247 869 303 342 376 385 431 623 394 128 795 406 268 449 657 236 206 724
 621 869 757 829 161 532 381 381 693 879 222 671 321 446 589 679 666 304
 864 142 356 800 130 620 606 268 650 253 871 154 203 857 207 512 665 298
 696 226 723 602 786 265 237 608 242 872 811 162 505 716 521 398 166 861
 336 866 265 551 824 136 782 733 400 104 412 690 518 540 508 217 802 203
 297 523 846 235 655 891 523 664 195 809]
i.  All odd numbers:
     Count:  48
     List:  [247, 869, 303, 385, 431, 623, 795, 449, 657, 621, 869, 757, 829, 161, 381, 381, 693, 879, 671,
321, 589, 679, 253, 871, 203, 857, 207, 665, 723, 265, 237, 811, 505, 521, 861, 265, 551, 733, 217, 203, 297
, 523, 235, 655, 891, 523, 195, 809]
ii. All odd numbers:
     Count:  52
     List:  [342, 376, 394, 128, 406, 268, 236, 206, 724, 532, 222, 446, 666, 304, 864, 142, 356, 800, 130,
620, 606, 268, 650, 154, 512, 298, 696, 226, 602, 786, 608, 242, 872, 162, 716, 398, 166, 336, 866, 824, 136
, 782, 400, 104, 412, 690, 518, 540, 508, 802, 846, 664]
iii. All prime numbers:
     Count:  11
     List:  [431, 449, 757, 829, 857, 811, 521, 733, 523, 523, 809]
```

7 Program 7

7.1 Question

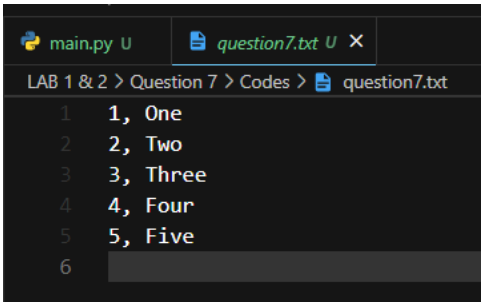
D is a dictionary defined as D=1:"One",2:"Two",3:"Three",4:"Four", 5:"Five". WAP to read all the keys and values from dictionary and write to the file in the given below format.
Key1, Value1
Key2, Value2
Key3, Value3

7.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


def writeToFile(filename, data):
    print('Writes the output to the file')
    file = open(filename, 'w+')
    for x in data:
        file.write(x)
        file.write('\n')
    print(data)
    file.close()


def main():
    D = {1: "One", 2: "Two", 3: "Three", 4: "Four", 5: "Five"}
    writeToFile('question7.txt', [(', '.join([str(x), str(D[x])])) for x in D])

if __name__ == '__main__':
    main()
```

7.3 Output

Question 7 Outputs



8 Program 8

8.1 Question

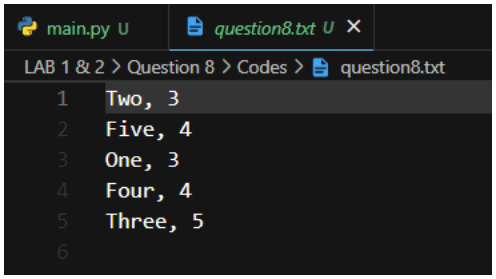
L is a list defined as L="One","Two","Three","Four","Five". WAP to count the length of reach element from a list and write to the file in the given below format:
One, 3
Two, 3
Four, 4

8.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


def writeToFile(filename, data):
    print('Writes the output to the file')
    file = open(filename, 'w+')
    for x in data:
        file.write(x)
        file.write('\n')
    print(data)
    file.close()


def main():
    L={"One","Two","Three","Four","Five"}
    writeToFile('question8.txt', [(', '.join([x, str(len(x))])) for x in L])

if __name__ == '__main__':
    main()
```

8.3 Output

Question 8 Outputs



9 Program 9

9.1 Question

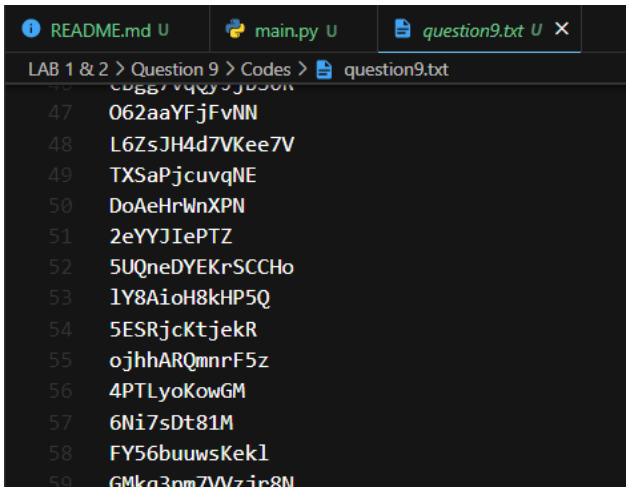
Write to the file 100 random strings whose length between 10 and 15.

9.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


import string
import numpy as np
import random


def writeToFile(filename, data):
    print('Writes the output to the file')
    file = open(filename, 'w+')
    for x in data:
        file.write(x)
    print(data)
    file.close()



def main():
    str = []
    for i in range(0, 100):
        print('Generating random strings')
        str.append(''.join(random.choices(string.ascii_letters +
                    string.digits, k=int(np.random.randint(10, 16, 1))))+'\n')
    print(str)
    writeToFile('question9.txt', str)

if __name__ == '__main__':
    main()
```

9.3 Output

Question 9 Outputs



10 Program 10

10.1 Question

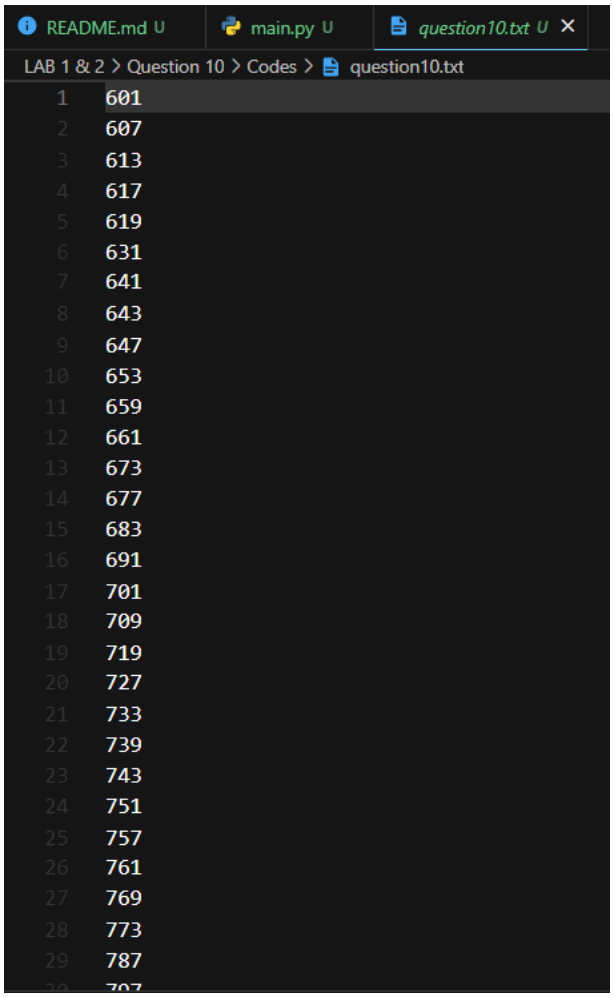
Write to the file all prime numbers between 600 and 800.

10.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'


import string
import numpy as np
import random


def writeToFile(filename, data):
    print('Writes the output to the file')
    file = open(filename, 'w+')
    for x in data:
        file.write(x)
    print(data)
    file.close()


def isPrime(x):
    if x > 1:
        for i in range(2, int(x**(1/2))+1):
            if x % i == 0:
                return False
        return True
    return False


def main():
    primes = []
    for i in range(600, 801):
        print('Finding prime numbers')
        if isPrime(i):
            primes.append(str(i)+'\n')
    print(primes)
    writeToFile('question10.txt', primes)


if __name__ == '__main__':
    main()
```

10.3 Output

Question 10 Outputs




```
601
607
613
617
619
631
641
643
647
653
659
661
673
677
683
691
701
709
719
727
733
739
743
751
757
761
769
773
787
```

11 Program 11

11.1 Question

WAP to calculate the time taken by a program.

11.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'

import time

def isPrime(x):
    if x>1:
        for i in range(2, int(x**(1/2))+1):
            if x % i == 0 : return False
        return True
    return False

def main():
    start = time.time()
    print('Prime numbers between 1 to 1000')
    for i in range(1, 1001):
        if isPrime(i):
            print(i, end=',')
    end = time.time()
    print('\nTime taken by the program: {}'.format(end-start))


if __name__ == '__main__':
    main()
```

11.3 Output

Question 11 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 11\Codes> python .\main.py
Prime numbers between 1 to 1000
2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97,101,103
,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,199,2
11,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,317
,331,337,347,349,353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,4
43,449,457,461,463,467,479,487,491,499,503,509,521,523,541,547,557,563,569,571
,577,587,593,599,601,607,613,617,619,631,641,643,647,653,659,661,673,677,683,6
91,701,709,719,727,733,739,743,751,757,761,769,773,787,797,809,811,821,823,827
,829,839,853,857,859,863,877,881,883,887,907,911,919,929,937,941,947,953,967,9
71,977,983,991,997,
Time taken by the program: 0.0009970664978027344
```

12 Program 12

12.1 Question

WAP to create a dictionary of student marks in five subjects and you have to find the student having maximum and minimum average marks.

12.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'

import numpy as np


def findAverage(data):
    avg = sum(data)/len(data)
    return avg


def main():
    D = {'Akshat': {'Physics': 95, 'Chemistry': 91, 'Maths': 99, 'Computer': 92, 'English': 88},
         'Raj': {'Physics': 80, 'Chemistry': 92, 'Maths': 100, 'Computer': 95, 'English': 87},
         'Vansh': {'Physics': 92, 'Chemistry': 95, 'Maths': 92, 'Computer': 99, 'English': 90}}
    print('Marks of each Student:')
    print(D)
    A ={}
    for i in range(0,len(D)):
        A[list(D.keys())[i]]=findAverage(list((list(D.values())[i]).values()))
    print('Average Marks of each Student:')
    print(A)
    minAvg = np.min(list(A.values()))
    maxAvg = np.max(list(A.values()))
    print('Minimum average marks is of: {}'.format(list(A.keys())[list(A.values()).index(minAvg)]))
    print('Maximum average marks is of: {}'.format(list(A.keys())[list(A.values()).index(maxAvg)]))
if __name__ == '__main__':
    main()
```

12.3 Output

Question 12 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 12\Codes> python .\main.py
Marks of each Student:
{'Akshat': {'Physics': 95, 'Chemistry': 91, 'Maths': 99, 'Computer': 92, 'English': 8
8}, 'Raj': {'Physics': 80, 'Chemistry': 92, 'Maths': 100, 'Computer': 95, 'English':
87}, 'Vansh': {'Physics': 92, 'Chemistry': 95, 'Maths': 92, 'Computer': 99, 'English'
: 90}}
Average Marks of each Student:
{'Akshat': 93.0, 'Raj': 90.8, 'Vansh': 93.6}
Minimum average marks is of: Raj
Maximum average marks is of: Vansh
```

13 Program 13

13.1 Question

WAP to sort the following number of elements in a list and calculate time taken.

13.2 Code

```python
__author__ = 'Akshat Raj Vansh'
__version__ = '0.1.0'
__license__ = 'MIT'

import random
import time

def timeToSort(size):
    L = list(range(size))
    random.shuffle(L)
    begin = time.time()
    L.sort()
    end = time.time()
    print('Time taken to sort a list of {s} elements is {t}'.format(s=size,t= end-begin))


def main():
    for i in range(5000,25001,5000):
        timeToSort(i)

if __name__ == '__main__':
    main()
```

13.3 Output

Question 13 Outputs

```
PS D:\Coding\Data Mining\LAB 1 & 2\Question 13\Codes> python .\main.py
Time taken to sort a list of 5000 elements is 0.0010004043579101562
Time taken to sort a list of 10000 elements is 0.0009996891021728516
Time taken to sort a list of 15000 elements is 0.0019686222076416016
Time taken to sort a list of 20000 elements is 0.0029931068420410156
Time taken to sort a list of 25000 elements is 0.004005908966064453
```