session-2-task-solutions

February 13, 2024

0.0.1 Problem 1: Write a program that will give you in hand monthly salary after deduction on CTC - HRA(10%), DA(5%), PF(3%) and taxes deduction as below:

Salary(Lakhs) : Tax(%)

- Below 5 : 0%
- 5-10 : 10%
- 10-20 : 20%
- aboove 20 : 30%

[ ]:

```python
# Write code here
#write your code
ctc = int(input('Enter your anual CTC:'))

if ctc < 5000000:
    salary=ctc*.82
elif ctc<1000000:
    salary=ctc*.72
elif ctc<2000000:
    salary=ctc*.62
else:
    salary=ctc*.52

print("You in hand monthly salary will be-", round(salary/12,2))
```

```
Enter your anual CTC:1200000
You in hand monthly salary will be- 82000.0
```

0.0.2 Problem 2: Write a program that take a user input of three angles and will find out whether it can form a triangle or not.

Hint - Sum of all angles is 180 and all angles are positive

[ ]:

```python
# Write code here
first = int(input('enter the 1st angle'))
second = int(input('enter the 2nd angle'))
third = int(input('enter the 3rd angle'))

if (first+second+third) == 180 and first>0 and second>0 and third>0:
  print('forms a triangle')
else:
  print('does not form a triangle')
```

```
enter the 1st angle30
enter the 2nd angle180
enter the 3rd angle-30
does not form a triangle
```

0.0.3 Problem 3: Write a program that will take user input of cost price and selling price and determines whether its a loss or a profit.

[ ]:

```
# Write code here
# write your code here
cost_price = int(input('Enter cost price-'))
selling_price = int(input('Enter selling price-'))

if cost_price < selling_price:
    print('Profit')
elif cost_price > selling_price:
    print('Loss')
else:
    print('No Loss No Gain')
```

```
Enter cost price-6000
Enter selling price-12000
Profit
```

0.0.4 Problem 4: Write a menu-driven program -

1. cm to ft
2. km to miles
3. USD to INR
4. exit

Hint - 1 cm = 0.032ft - 1km = 0.62 - 1 USD = 80 INR

[ ]:

```
# Write code here
menu = input("""
Hi select an option
1. cms to ft
2. km to miles
3. USD to INR
4. Exit
""")

if menu == '1':
  cm = float(input('enter the cm value'))
  print('ft value is',0.032*cm)
elif menu == '2':
  km = float(input('enter the km value'))
  print('miles value is',km*0.62)
elif menu == '3':
  usd = float(input('enter usd'))
  print('inr',usd*80)
else:
  exit()
```

```
Hi select an option
1. cms to ft
2. km to miles
3. USD to INR
4. Exit
4
```

0.0.5 `Problem 5` - Exercise 12: Display Fibonacci series up to 10 terms.

Note: The Fibonacci Sequence is a series of numbers. The next number is found by adding up the two numbers before it. The first two numbers are 0 and 1. For example, 0, 1, 1, 2, 3, 5, 8, 13, 21. The next number in this series above is 13+21 = 34

[ ]:

```
# Write code here

num1,num2 = 0,1

for i in range(10):
  print(num1)

  next = num1 + num2

  num1 = num2
  num2 = next
```

```
0
1
1
2
3
5
8
13
21
34
```

0.0.6 Problem 6 - Find the factorial of a given number.

Write a program to use the loop to find the factorial of a given number.

The factorial (symbol: `!`) means to multiply all whole numbers from the chosen number down to 1.

For example: calculate the factorial of 5

`5! = 5 × 4 × 3 × 2 × 1 = 120`

Output:

`120`

[ ]:

```
# Write code here
num = int(input('enter the number'))

fact = 1
for i in range(1,num+1):
  fact = fact*i

print(fact)
```

```
enter the number5
120
```

0.0.7 Problem 7 - Reverse a given integer number.

Example:

`Input:`

`76542`

`Output:`

`24567`

[ ]:

```
# Write code here
number = int(input('enter the number'))

rev = 0

while number>0:
  last = number%10
  rev = rev*10 + last
  number = number//10

print(rev)
```

```
enter the number123456
654321
```

0.0.8 Problem 8: Take a user input as integer N. Find out the sum from 1 to N. If any number if divisible by 5, then skip that number. And if the sum is greater than 300, don't need to calculate the sum further more. Print the final result. And don't use for loop to solve this problem.

Example 1:

Input:

30

Output:

276

[ ]:

```
# Write code here
N = int(input('enter the number'))
sum = 0
i = 1

while i < N+1:
  if i % 5 == 0:
    i+=1
    continue

  sum += i

  if sum > 300:
    sum = sum - i
    break

  i+=1

print(sum)
```

```
enter the number50
276
```

0.0.9 Problem 9: Write a program that keeps on accepting a number from the user until the user enters Zero. Display the sum and average of all the numbers.

[ ]:

```
# Write code here
sum = 0
count = 0

while True:
  num = int(input('enter number'))
  if num == 0:
    break
  sum = sum + num
  count = count + 1

print('sum',sum)
print('avg',sum/count)
```

```
enter number1
enter number2
enter number3
enter number4
enter number5
enter number0
sum 15
avg 3.0
```

###Problem 9: Write a program 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 printed in a comma-separated sequence on a single line.

[ ]:

```
# Write code here
L = []
for i in range(2000,3201):
  if i % 7 == 0 and i % 5 != 0:
    L.append(str(i))

print(",".join(L))
```

```
2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,
2142,2149,2156,2163,2177,2184,2191,2198,2212,2219,2226,2233,2247,2254,2261,2268,
2282,2289,2296,2303,2317,2324,2331,2338,2352,2359,2366,2373,2387,2394,2401,2408,
2422,2429,2436,2443,2457,2464,2471,2478,2492,2499,2506,2513,2527,2534,2541,2548,
2562,2569,2576,2583,2597,2604,2611,2618,2632,2639,2646,2653,2667,2674,2681,2688,
2702,2709,2716,2723,2737,2744,2751,2758,2772,2779,2786,2793,2807,2814,2821,2828,
2842,2849,2856,2863,2877,2884,2891,2898,2912,2919,2926,2933,2947,2954,2961,2968,
2982,2989,2996,3003,3017,3024,3031,3038,3052,3059,3066,3073,3087,3094,3101,3108,
3122,3129,3136,3143,3157,3164,3171,3178,3192,3199
```

###Problem 10: Write a program, 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 printed in a space-separated sequence on a single line.

[ ]:

```
# Write code here
L = []
for i in range(1000,3001):
  flag = True

  curr = i

  while curr > 0:
    last = curr%10
    if last % 2 != 0:
      flag = False
      break
    curr = curr//10

  if flag == True:
    L.append(str(i))

print(",".join(L))
```

```
2000,2002,2004,2006,2008,2020,2022,2024,2026,2028,2040,2042,2044,2046,2048,2060,
2062,2064,2066,2068,2080,2082,2084,2086,2088,2200,2202,2204,2206,2208,2220,2222,
2224,2226,2228,2240,2242,2244,2246,2248,2260,2262,2264,2266,2268,2280,2282,2284,
2286,2288,2400,2402,2404,2406,2408,2420,2422,2424,2426,2428,2440,2442,2444,2446,
2448,2460,2462,2464,2466,2468,2480,2482,2484,2486,2488,2600,2602,2604,2606,2608,
2620,2622,2624,2626,2628,2640,2642,2644,2646,2648,2660,2662,2664,2666,2668,2680,
2682,2684,2686,2688,2800,2802,2804,2806,2808,2820,2822,2824,2826,2828,2840,2842,
2844,2846,2848,2860,2862,2864,2866,2868,2880,2882,2884,2886,2888
```

###Problem 11: 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.
>
> ! means robot stop there.

Please write a program to compute the distance from current position after a sequence of movement and original point.

If the distance is a float, then just print the nearest integer.

Example:

Input:

```
UP 5
DOWN 3
LEFT 3
RIGHT 2
!
```

Output:

```
2
```

[ ]:

```
# Write code here
pos = [0,0]
while True:
  s = input('Enter the robot path')
  if s == '!':
    break
  direction = s.split()[0]
  steps = int(s.split()[1])

  if direction == 'UP':
    pos[1] = pos[1] + steps
  elif direction == 'DOWN':
    pos[1] = pos[1] - steps
  elif direction == 'LEFT':
    pos[0] = pos[0] - steps
  elif direction == 'RIGHT':
    pos[0] = pos[0] + steps
  else:
    pass

print('new pos',pos)
print((pos[0]**2 + pos[1]**2)**0.5)
```

```
Enter the robot pathUP 10
Enter the robot pathDOWN 5
Enter the robot pathLEFT 13
Enter the robot path!
new pos [-13, 5]
13.92838827718412
```

###Problem 12:Write a program to print whether a given number is a prime number or not

[ ]:

```
# Write code here
num = int(input('enter the num'))

flag = True
for i in range(2,num):
  if num%i == 0:
    flag = False
    break

if flag == True:
  print('Prime')
else:
  print('Not Prime')
```

```
enter the num21
Not Prime
```

###Problem 13:Print all the Armstrong numbers in a given range. Range will be provided by the user Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

[ ]:
```
# Write code here
start = int(input("Enter the start of the range: "))
end = int(input("Enter the end of the range: "))

for num in range(start, end+1):
    temp = num
    sum = 0
    order = len(str(num))
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10
    if num == sum:
        print(num)
```

```
Enter the start of the range: 0
Enter the end of the range: 200
0
1
2
3
4
5
6
7
8
9
153
```

###Problem 14:Calculate the angle between the hour hand and minute hand.

Note: There can be two angles between hands; we need to print a minimum of two. Also, we need to print the floor of the final result angle. For example, if the final angle is 10.61, we need to print 10.

Input: H = 9 , M = 0 Output: 90 Explanation: The minimum angle between hour and minute hand when the time is 9 is 90 degress.

[ ]:
```
# Write code here

# Write code here
h = int(input('Enter hours hand-'))
m = int(input('Enter minute hand-'))

# validate the input
```

```python
if (h < 0 or m < 0 or h > 12 or m > 60):
    print('Wrong input')

# Idea is to minute angle - hour agnle from clockwise from 12 hour point

# 1 minute in minute angle manke 6 degree. (60 minute -> 360 degree)
m_angle = m*6

# every hour point yeilds to 30degree-- 12 hours-360degree plus if minute hand␣
 ↪moves hour hands move too
# Every minute after hour hand take 0.5 degree movement. clockwise
h_angle = h*30 + m*0.5

# Take abs difference b/w them
angle = abs(h_angle - m_angle)


if angle>180:
    print(360-angle)
else:
    print(angle)
```

```
Enter hours hand-12
Enter minute hand-00
0.0
```

Explaination: This code calculates the angle between the hour and minute hand on a clock. It takes user input for the hour and minute hand positions, validates the input to make sure it is within the range of a clock (0-12 for hours and 0-60 for minutes), then calculates the angles of the hour and minute hand based on their position on the clock. It then finds the difference between the two angles and takes the absolute value of that difference. It then checks if the angle is greater than 180, if it is, it prints the difference of 360 and the angle, else it prints the angle. The final output is the angle between the hands on the clock.

###Problem 15:Given two rectangles, find if the given two rectangles overlap or not. A rectangle is denoted by providing the x and y coordinates of two points: the left top corner and the right bottom corner of the rectangle. Two rectangles sharing a side are considered overlapping. (L1 and R1 are the extreme points of the first rectangle and L2 and R2 are the extreme points of the second rectangle).

Note: It may be assumed that the rectangles are parallel to the coordinate axis.

[ ]:

```python
# Write code here
print("Enter both Rectangel points coordinates <x y>")

l1_x, l1_y = list(map(int, input('Enter L1 coordinate value  (Two value space␣
 ↪seperated):').split()))
r1_x, r1_y = list(map(int, input('Enter r1 coordinate value,(Two value space␣
 ↪seperated):').split()))
```

```
l2_x, l2_y = list(map(int, input('Enter L2 coordinate value,(Two value space␣
 ↪seperated):').split()))
r2_x, r2_y = list(map(int, input('Enter r2 coordinate value,(Two value space␣
 ↪seperated):').split()))

# if rectangle has area 0, no overlap
if l1_x == r1_x or l1_y == r1_y or r2_x == l2_x or l2_y == r2_y:
  print("Don't Overlap")

# If one rectangle is on left side of other
elif l1_x > r2_x or l2_x > r1_x:
  print("Don't Overlap")

# If one rectangle is above other
elif r1_y > l2_y or r2_y > l1_y:
  print("Don't Overlap")
else:
  print("Overlap")
```

`Explaination:` The given code takes the coordinates of two rectangles from the user and checks if they overlap. * It does this by first checking if either rectangle has an area of 0 (i.e. one of the sides has length 0), in which case it prints "Don't Overlap". * It then checks if one rectangle is entirely to the left or above the other, in which case it also prints "Don't Overlap". If none of these conditions are true, it prints "Overlap".

session-3-task-solutions

February 13, 2024

0.0.1 Problem 1 - Print the following pattern. Write a program to use for loop to print the following reverse number pattern.

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

[ ]:

```
# Code here

rows = int(input('enter the rows'))

for i in range(0,rows):
  for j in range(rows-i,0,-1):
    print(j,end=' ')
  print()
```

```
enter the rows5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

0.0.2 Problem 2: Print the following pattern.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

[ ]:

```
# Code here
rows = int(input('enter the rows'))
```

```
for i in range(1,rows+1):
  for j in range(0,i):
    print('*',end=' ')
  print()

for i in range(1,rows):
  for j in range(rows-i,0,-1):
    print('*',end=' ')
  print()
```

```
enter the rows5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

0.0.3 `Problem` 3:Write a program to pring the following pattern

```
    *
   * *
  * * *
```

[ ]:

```
# Code here
rows = 6
for i in range(1,rows+1):
  print(' '*rows,end='')
  print('* '*i)
  rows = rows - 1
```

```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *
```

0.0.4 `Problem` 4:Write a program to print the following pattern

1

2 1

3 2 1

4 3 2 1

5 4 3 2 1

[ ]:

```
# Code here
rows = 5

for i in range(1,rows+1):
  for j in range(i,0,-1):
    print(j,end=' ')
  print()
```

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

0.0.5 Problem 5: Write a Python Program to Find the Sum of the Series till the nth term:

1 + x^2/2 + x^3/3 + … x^n/n n will be provided by the user

[ ]:

```
# Code here
x = 10
n = 5

sum = 1
s = ''
print('1 + ',end='')
for i in range(2,n+1):
  sum = sum + x**i/i
  s = s + 'x^{}/{} +'.format(i,i)
print(s[:-1])
print(sum)
```

```
1 + x^2/2 +x^3/3 +x^4/4 +x^5/5
22884.333333333332
```

0.0.6 Problem 6: The natural logarithm can be approximated by the following series.

If x is input through the keyboard, write a program to calculate the sum of the first seven terms of this series.

[ ]:

```
# Code here
x = 10
n = 5

sum = 0
s = ''
for i in range(1,n+1):
  sum = sum + (1/i)*((x-1)/x)**i
  s = s + '(1/{})*((x-1)/x)^{} +'.format(i,i)
print(s[:-1])
print(sum)
```

```
(1/1)*((x-1)/x)^1 +(1/2)*((x-1)/x)^2 +(1/3)*((x-1)/x)^3 +(1/4)*((x-1)/x)^4
+(1/5)*((x-1)/x)^5
1.8301230000000004
```

0.0.7 Problem 7 - Find the sum of the series upto n terms.

Write a program to calculate the sum of series up to n term. For example, if n =5 the series will become 2 + 22 + 222 + 2222 + 22222 = 24690. Take the user input and then calculate. And the output style should match which is given in the example.

Example 1:

Input:

5

Output:

```
2+22+222+2222+22222
Sum of above series is: 24690
```

[ ]:

```
# Code here
n = int(input('enter the number of terms'))

start = 2
sum = 0

for i in range(0,n):
  if i < n-1:
    print(start,end='+')
  else:
    print(start)

  sum = sum + start
  start = start*10 + 2

print(sum)
```

```
enter the number of terms5
2+22+222+2222+22222
24690
```

###Problem 8: Write a program to print all the unique combinations of 1,2,3 and 4

Output:

```
1 2 3 4
1 2 4 3
1 3 2 4
1 3 4 2
1 4 2 3
1 4 3 2
2 1 3 4
2 1 4 3
2 3 1 4
2 3 4 1
2 4 1 3
.
.
and so on
```

[ ]:

```python
# Code here
for i in range(1,5):
  for j in range(1,5):
    for k in range(1,5):
      for m in range(1,5):
        print(i,j,k,m)
```

###Problem 9: Write a program that will take a decimal number as input and prints out the binary equivalent of the number

[ ]:

```python
# Code here

n = int(input('enter the number'))
binary = []
while n > 0:

  binary.append(n%2)
  n = n//2

for i in binary[::-1]:
  print(i,end='')
```

```
enter the number3
11
```

###Problem 10: Write a program that will take 2 numbers as input and prints the LCM and HCF of those 2 numbers

[ ]:

```
# Code here
x = int(input('enter 1st number'))
y = int(input('enter 2nd number'))

if x>y:
  greater = x
else:
  greater = y

while True:
  if (greater % x == 0) and (greater % y == 0):
    lcm = greater
    break

  greater = greater + 1

print(lcm)
```

```
enter 1st number10
enter 2nd number15
30
```

[ ]:

```
# Code here
x = int(input('enter 1st number'))
y = int(input('enter 2nd number'))

if x<y:
  smaller = x
else:
  smaller = y

for i in range(1,smaller+1):
  if (x % i == 0) and (y % i == 0):
    hcf = i

print(hcf)
```

```
enter 1st number12
enter 2nd number16
4
```

0.0.8 Problem 11: Create Short Form from initial character

Given a string create short form ofthe string from Initial character. Short form should be capitalised.

Example:

```
Input:
```

```
Data science mentorship program
```

```
Output:
```

```
DSMP
```

[ ]:

```python
# Code here

inp = 'Data science Mentorship Program'

res = ''

for i in inp.split():
  res = res + i[0].upper()

print(res)
```

```
DSMP
```

###Problem 12: Append second string in the middle of first string

```
Input:
```

```
campusx
data
```

```
Output:
```

```
camdatapusx
```

[ ]:

```python
# Code here

s1 = input('enter the 1st string')
s2 = input('enter the 2nd string')

print(s1[0:int(len(s1)/2)] + s2 + s1[int(len(s1)/2):])
```

```
enter the 1st stringwill
enter the 2nd stringdo
widoll
```

0.0.9 Problem 13:Given string contains a combination of the lower and upper case letters. Write a program to arrange the characters of a string so that all lowercase letters should come first.

Given:

```
str1 = PyNaTive
```

Expected Output:

```
yaivePNT
```

[ ]:

```python
# Code here

s = 'PyNaTive'

upper = ''
lower = ''

for i in s:
  if i.islower():
    lower = lower + i
  else:
    upper = upper + i

print(lower + upper)
```

```
yaivePNT
```

0.0.10 Problem 14:Take a alphanumeric string input and print the sum and average of the digits that appear in the string, ignoring all other characters.

Input:

```
hel12304every093
```

Output:

```
Sum: 22
Avg: 2.75
```

[ ]:

```python
# Code here
s = 'hel12304every093'

sum = 0
count = 0

for i in s:
  if i.isdigit():
    sum = sum + int(i)
    count += 1

print(sum)
print(sum/count)
print(count)
```

```
22
3.142857142857143
7
```

0.0.11 Problem 15: Removal of all characters from a string except integers

Given:

str1 = 'I am 25 years and 10 months old'

Expected Output:

2510

[ ]:

```
# Code here
s = 'I am 25 years and 10 months old'

res = ''

for i in s:
  if i.isdigit():
    res = res + i

print(res)
```

```
2510
```

0.0.12 Problem 16: Check whether the string is Symmetrical.

Statement: Given a string. the task is to check if the string is symmetrical or not. A string is said to be symmetrical if both the halves of the string are the same.

Example 1:

Input

khokho

Output

The entered string is symmetrical

[ ]:

```
# Code here
s = input('enter the string')

if len(s)%2 == 0:
  s1 = s[0:len(s)//2]
  s2 = s[len(s)//2:]
else:
  s1 = s[0:len(s)//2]
  s2 = s[len(s)//2 + 1:]

if s1 == s2:
  print('symmetrical')
else:
  print('not symmetrical')
```

```
enter the stringmadam
not symmetrical
```

0.0.13 `Problem 17:` Reverse words in a given String

Statement: We are given a string and we need to reverse words of a given string.

Example 1:

```
Input:
```

```
geeks quiz practice code
```

```
Output:
```

```
code practice quiz geeks
```

Example 2:

```
Input:
```

```
my name is laxmi
```

```
Output:
```

```
laxmi is name my
```

[ ]:

```python
# Code here
s = 'code practice quiz geeks'

L = []

for i in s.split():
  L.append(i)

L = L[::-1]
print(" ".join(L))
```

```
geeks quiz practice code
```

0.0.14 `Problem 18:` Find uncommon words from two Strings.

Statement: Given two sentences as strings **A** and **B**. The task is to return a list of all uncommon words. A word is uncommon if it appears exactly once in any one of the sentences, and does not appear in the other sentence. Note: A sentence is a string of space-separated words. Each word consists only of lowercase letters.

Example 1:

```
Input:
```

```
A = "apple banana mango"
B = "banana fruits mango"
```

```
Output:
```

```
['apple', 'fruits']
```

[ ]:
```python
# Code here
A = "apple banana mango"
B = "banana fruits mango"

L = []

for i in A.split():
  if i not in B and i not in L:
    L.append(i)

for j in B.split():
  if j not in A and j not in L:
    L.append(j)

print(L)
```

```
['apple', 'fruits']
```

0.0.15 Problem 19: Word location in String.

Statement: Find a location of a word in a given sentence.

Example 1:

Input:

Sentence: We can learn data science through campusx mentorship program.

word: campusx

Output:

Location of the word is 7.

Note- Don't use index/find functions

[ ]:
```python
# Code here
s = 'We can learn data science through campusx mentorship program.'
word = 'campusx'

pos = 0

for i in s.split():
  pos += 1
  if i == word:
    break

print(pos)
```

```
7
```

0.0.16 Problem 20: Write a program that can remove all the duplicate characters from a string. User will provide the input.

[ ]:

```python
# Code here
s = 'aaaabbbbbccccdddeeeeffff'

res = ''

for i in s:
  if i not in res:
    res = res + i

print(res)
```

```
abcdef
```

[ ]:

session-4-task-solutions

February 13, 2024

###`Problem 1:` Combine two lists index-wise(columns wise)

Write a program to add two lists index-wise. Create a new list that contains the 0th index item from both the list, then the 1st index item, and so on till the last element. any leftover items will get added at the end of the new list.

`Given List:`

```
list1 = ["M", "na", "i", "Kh"]
list2 = ["y", "me", "s", "an"]
```

`Output:`

```
[['M','y'], ['na', me'], ['i', 's'], ['Kh', 'an']]
```

[28]:
```python
# Write code here
list1 = ["M", "na", "i", "Kh"]
list2 = ["y", "me", "s", "an"]

[[i,j] for (i,j) in zip(list1,list2)]
```

[28]:
```
[['M', 'y'], ['na', 'me'], ['i', 's'], ['Kh', 'an']]
```

0.0.1 `Problem 2:` Add new item to list after a specified item

Write a program to add item 7000 after 6000 in the following Python List

`list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]`

`Output:`

```
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
```

[29]:
```python
# Write code here
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]
list1[2][2].append(7000)
list1
```

[29]:
```
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
```

###`Problem 3:` Update no of items available

Suppose you are given a list of candy and another list of same size representing no of items of respective candy.

i.e -

```
candy_list = ['Jelly Belly','Kit Kat','Double Bubble','Milky Way','Three Musketeers']
no_of_items = [10,20,34,74,32]
```

Write a program to show no. of items of each candy type.

`Output:`

```
Jelly Belly-10
Kit Kat-20
Double Bubble-34
Milky Way-74
Three Musketeers-32
```

[30]:
```
# Write code here

candy_list = ['Jelly Belly','Kit Kat','Double Bubble','Milky Way','Three␣
  ↪Musketeers']
no_of_items = [10,20,34,74,32]

for (i,j) in zip(candy_list,no_of_items):
  print(i,'-',j)
```

```
Jelly Belly - 10
Kit Kat - 20
Double Bubble - 34
Milky Way - 74
Three Musketeers - 32
```

###Problem 4: Running Sum on list Write a program to print a list after performing running sum on it.

i.e:

`Input:`

`list1 = [1,2,3,4,5,6]`

`Output:`

`[1,3,6,10,15,21]`

[31]:
```
# Write code here
list1 = [1,2,3,4,5,6]

result = []
sum = 0

for i in list1:
  sum = sum + i
  result.append(sum)

print(result)
```

```
[1, 3, 6, 10, 15, 21]
```

###Problem 5: You are given a list of integers. You are asked to make a list by running through elements of the list by adding all elements greater and itself.

i.e. Say given list is `[2,4,6,10,1]` resultant list will be `[22,20,10,23]`.

For 1st element `2` ->> these are greater `(4+6+10)` values and `2` itself so on adding becomes `22`.

For 2nd element `4` ->> greater elements are `(6, 10)` and `4` itself, so on adding `20`

like wise for all other elememts.

`[2,4,6,10,1]`->`[22,20,16,10,23]`

[32]:

```
# Write code here
L = [2,4,6,10,1]
result = []

for i in L:
  sum = 0
  for j in L:
    if i <= j:
      sum = sum + j

  result.append(sum)

print(result)
```

```
[22, 20, 16, 10, 23]
```

###Problem 6: Find list of common unique items from two list. and show in increasing order

`Input`

```
num1 = [23,45,67,78,89,34]
num2 = [34,89,55,56,39,67]
```

`Output:`

`[34, 67, 89]`

[33]:

```
# Write code here
num1 = [23,45,67,78,89,34,67]
num2 = [34,89,55,56,39,67,67]

common = []
```

```
for i in num1:
  if i in num2:
    if i not in common:
      common.append(i)

print(common)
```

```
[67, 89, 34]
```

###Problem 7: Sort a list of alphanumeric strings based on product value of numeric character in it. If in any string there is no numeric character take it's product value as 1.

Input:

['1ac21', '23fg', '456', '098d','1','kls']

Output:

['456', '23fg', '1ac21', '1', 'kls', '098d']

[34]:

```
# Write code here
L = ['1ac21', '23fg', '456', '098d','1','kls']
prod_value = []
for item in L:
  product = 1
  for char in item:
    if char.isdigit():
      product = product*int(char)

  prod_value.append(product)

[i[1] for i in sorted(list(zip(prod_value,L)),reverse=True)]
```

[34]:

```
['456', '23fg', '1ac21', 'kls', '1', '098d']
```

Problem 8: Split String of list on K character.

Example :

Input:

['CampusX is a channel', 'for data-science', 'aspirants.']

Output:

['CampusX', 'is', 'a', 'channel', 'for', 'data-science', 'aspirants.']

[35]:

```
# Write code here
L = ['CampusX is a channel', 'for data-science', 'aspirants.']
inp = 'a'
result = []
for i in L:
  result.extend(i.split(inp))
```

```
print(result)
```

```
['C', 'mpusX is ', ' ch', 'nnel', 'for d', 't', '-science', '', 'spir', 'nts.']
```

0.0.2 Problem 9: Convert Character Matrix to single String using string comprehension.

Example 1:

Input:

```
[['c', 'a', 'm', 'p', 'u', 'x'], ['i', 's'], ['b', 'e', 's', 't'], ['c', 'h', 'a', 'n', 'n', '
```

Output:

```
campux is best channel
```

[36]:

```
# Write code here
L = [['c', 'a', 'm', 'p', 'u', 'x'], ['i', 's'], ['b', 'e', 's', 't'], ['c',␣
 ↪'h', 'a', 'n', 'n', 'e', 'l']]

print(" ".join(["".join(i) for i in L]))
```

```
campux is best channel
```

0.0.3 Problem 10: Add Space between Potential Words.

Example:

Input:

```
['campusxIs', 'bestFor', 'dataScientist']
```

Output:

```
['campusx Is', 'best For', 'data Scientist']
```

[37]:

```
# Code here
# initializing list
test_list = ["campusxIs", "bestFor", "dataScientist"]

res = []

for i in test_list:
    temp = [[]]
    for char in i:

        if char.isupper():
            temp.append([])

        temp[-1].append(char)
    temp_string = ""
    for item in temp:
        temp_string = temp_string + "".join(item) + " "
    res.append(temp_string[0:-1])

print(res)
```

```
['campusx Is', 'best For', 'data Scientist']
```

0.0.4 Problem 11: Write a program that can perform union operation on 2 lists

Example:

Input:

```
[1,2,3,4,5,1]
[2,3,5,7,8]
```

Output:

```
[1,2,3,4,5,7,8]
```

[38]:

```
# Write code here
L1 = [1,2,3,4,5,1]
L2 = [2,3,5,7,8]

union = []

for i in L1:
  if i not in union:
    union.append(i)

for j in L2:
  if j not in union:
    union.append(j)

print(union)
```

```
[1, 2, 3, 4, 5, 7, 8]
```

0.0.5 Problem 12: Write a program that can find the max number of each row of a matrix

Example:

Input:

```
[[1,2,3],[4,5,6],[7,8,9]]
```

Output:

```
[3,6,9]
```

[39]:

```
# Write code here
L = [[1,2,3],[4,5,6],[7,8,9]]
result = []
for i in L:
  result.append(max(i))

print(result)
```

```
[3, 6, 9]
```

0.0.6 Problem 13: Write a list comprehension to print the following matrix

[[0, 1, 2], [3, 4, 5], [6, 7, 8]]

[40]:

```
# Write code here
[[j + 3*i for j in range(0,3)] for i in range(0,3)]
```

[40]:

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

0.0.7 Problem 14: Write a list comprehension that can transpose a given matrix

matrix = [[1,2,3], [4,5,6], [7,8,9]]

[1, 4, 7] [2, 5, 8] [3, 6, 9]

[41]:

```
# Write code here
matrix = [
[1,2,3],
[4,5,6],
[7,8,9]
]

[[row[i] for row in matrix]for i in range(len(matrix))]
```

[41]:

```
[[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

0.0.8 Problem 15: Write a list comprehension that can flatten a nested list

Input matrix = [[1,2,3], [4,5,6], [7,8,9]]

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]

[42]:

```
# Write code here
matrix = [
[1,2,3],
[4,5,6],
[7,8,9]
]
```

```


[item for row in matrix for item in row]
```

```
[42]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[42]:
```

session-5-task-solutions

February 13, 2024

1 Tuple

###Q1: Join Tuples if similar initial element While working with Python tuples, we can have a problem in which we need to perform concatenation of records from the similarity of initial element. This problem can have applications in data domains such as Data Science.

For eg.

```
Input  : test_list = [(5, 6), (5, 7), (5, 8), (6, 10), (7, 13)]
Output : [(5, 6, 7, 8), (6, 10), (7, 13)]
```

[ ]:

```
# write your code here
test_list = [(5, 6), (5, 7), (5, 8), (6, 10), (7, 13)]

unique = []

for i in test_list:
  unique.append(i[0])
unique = set(unique)

result = []
for i in unique:
  result.append([i])
  for j in test_list:
    if j[0] == i:
      result[-1].append(j[1])

print(list(map(tuple,result)))
```

```
[(5, 6, 7, 8), (6, 10), (7, 13)]
```

###Q2: Multiply Adjacent elements (both side) and take sum of right and lest side multiplication result.

For eg.

```
The original tuple : (1, 5, 7, 8, 10)
Resultant tuple after multiplication :

(1*5, 1*5+5*7, 7*5 + 7*8, 8*7 + 8*10, 10*8) -> (5, 40, 91, 136, 80)
```

```
output-(5, 40, 91, 136, 80)
```

[ ]:

```
# write your code here
t = (1, 5, 7, 8, 10)

L = []

L.append(t[0]*t[1])

for i in range(1,len(t)-1):
  L.append(t[i]*t[i-1] + t[i]*t[i+1])

L.append(t[-1]*t[-2])

print(tuple(L))
```

```
(5, 40, 91, 136, 80)
```

###Q3: Check is tuples are same or not? Two tuples would be same if both tuples have same element at same index

```
t1 = (1,2,3,0)
t2 = (0,1,2,3)

t1 and t2 are not same
```

[ ]:

```
# write your code here
t1 = (1,2,3,0)
t2 = (1,2,3,0)

flag = True
for i,j in zip(t1,t2):
  if i == j:
    continue
  else:
    flag = False
    break
if flag:
  print('same')
else:
  print('not same')
```

```
same
```

###Q4: Count no of tuples, list and set from a list

```
list1 = [{'hi', 'bye'},{'Geeks', 'forGeeks'},('a', 'b'),['hi', 'bye'],['a', 'b']]
```

```
Output:
```

```
List-2
```

```
Set-2
Tuples-1
```

[ ]:

```python
# write your code here
L = [{'hi', 'bye'},{'Geeks', 'forGeeks'},('a', 'b'),['hi', 'bye'],['a', 'b']]
output = [0,0,0]

for i in L:
  if type(i) == list:
    output[0] = output[0] + 1
  elif type(i) == set:
    output[1] = output[1] + 1
  elif type(i) == tuple:
    output[2] = output[2] + 1
  else:
    pass

print('Lists-{}\nSets-{}\nTuples-{}'.format(output[0],output[1],output[2]))
```

```
Lists-2
Sets-2
Tuples-1
```

###Q5: Shortlist Students for a Job role Ask user to input students record and store in tuples for each record. Then Ask user to input three things he wants in the candidate- Primary Skill, Higher Education, Year of Graduation.

Show every students record in form of tuples if matches all required criteria.

It is assumed that there will be only one primry skill.

If no such candidate found, print `No such candidate`

`Input:`

```
Enter No of records- 2
Enter Details of student-1
Enter Student name- Manohar
Enter Higher Education- B.Tech
Enter Primary Skill- Python
Enter Year of Graduation- 2022
Enter Details of student-2
Enter Student name- Ponian
Enter Higher Education- B.Sc.
Enter Primary Skill- C++
Enter Year of Graduation- 2020

Enter Job Role Requirement
Enter Skill- Python
Enter Higher Education- B.Tech
Enter Year of Graduation- 2022
```

```
Output
```

```
('Manohar', 'B.tech', 'Python', '2022')
```

[ ]:

```
# write your code here
students = []

num = int(input('enter the number of applicants'))

for i in range(num):
  print('Enter details of',i+1,'applicant:')
  name = input('enter name')
  h_ed = input('enter higher education')
  p_skill = input('enter primary skill')
  yog = input('enter year of graduation')

  students.append((name,h_ed,p_skill,yog))

required_skill = input('enter required skill')
required_hed = input('enter required higher education')
required_yog = input('enter required year of graduation')

flag = False
for i in students:
  if i[1] == required_hed and i[2] == required_skill and i[3] == required_yog:
    print(i)
    flag = True

if flag == False:
  print('No such candidates')
```

```
enter the number of applicants2
Enter details of 1 applicant:
enter nameNitish
enter higher educationB.Tech
enter primary skillPython
enter year of graduation2013
Enter details of 2 applicant:
enter nameAnkit
enter higher educationB.A
enter primary skillC++
enter year of graduation2016
enter required skillPython
enter required higher educationB.Tech
enter required year of graduation2013
('Nitish', 'B.Tech', 'Python', '2013')
```

2 Set

###Q1: Write a program to find set of common elements in three lists using sets.

```
Input : ar1 = [1, 5, 10, 20, 40, 80]
        ar2 = [6, 7, 20, 80, 100]
        ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

Output : [80, 20]
```

[ ]:

```python
# write your code here
ar1 = [1, 5, 10, 20, 40, 80]
ar2 = [6, 7, 20, 80, 100]
ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

s1 = set(ar1)
s2 = set(ar2)
s3 = set(ar3)

result = list((s1 & s2) & s3)
print(result)
```

```
[80, 20]
```

###Q2: Write a program to count unique number of vowels using sets in a given string. Lowercase and upercase vowels will be taken as different.

```
Input:

Str1 = "hands-on data science mentorship progrAm with live classes at affordable fee only on Ca

Output:

No of unique vowels-6
```

[ ]:

```python
# write your code here
vowels = set('aeiouAEIOU')

s = set("hands-on data science mentorship progrAm with live classes at␣
 ↪affordable fee only on CampusX")

print('No of unique vowels-',len(s & vowels))
```

```
No of unique vowels- 6
```

2.0.1 Q3: Write a program to Check if a given string is binary string of or not.

A string is said to be binary if it's consists of only two unique characters.

Take string input from user.

```
Input: str = "01010101010"
Output: Yes
```

```
Input: str = "1222211"
Output: Yes

Input: str = "Campusx"
Output: No
```

[ ]:

```
# write your code here
s = "010101010103"

if len(set(s)) == 2:
  print('binary')
else:
  print('not binary')
```

```
not binary
```

2.0.2 Q4: find union of n arrays.

Example 1:

Input:

```
[[1, 2, 2, 4, 3, 6],
 [5, 1, 3, 4],
 [9, 5, 7, 1],
 [2, 4, 1, 3]]
```

Output:

```
[1, 2, 3, 4, 5, 6, 7, 9]
```

[ ]:

```
# write your code here
L = [[1, 2, 2, 4, 3, 6],
 [5, 1, 3, 4],
 [9, 5, 7, 1],
 [2, 4, 1, 3]]

s = set()

for i in L:
  s.update(i)

print(s)
```

```
{1, 2, 3, 4, 5, 6, 7, 9}
```

2.0.3 Q5: Intersection of two lists. Intersection of two list means we need to take all those elements which are common to both of the initial lists and store them into another list. Only use using list-comprehension.

Example 1:

Input:

```
lst1 = {15, 9, 10, 56, 23, 78, 5, 4, 9}
lst2 = {9, 4, 5, 36, 47, 26, 10, 45, 87}
```

Output:

```
[9, 10, 4, 5]
```

Example 2:

Input:

```
lst1 = {4, 9, 1, 17, 11, 26, 28, 54, 69}
lst2 = {9, 9, 74, 21, 45, 11, 63, 28, 26}
```

Output:

```
[9, 11, 26, 28]
```

[ ]:

```
# write your code here
lst1 = {15, 9, 10, 56, 23, 78, 5, 4, 9}
lst2 = {9, 4, 5, 36, 47, 26, 10, 45, 87}

[item for item in lst1 if item in lst2]
```

[ ]:

```
[4, 5, 9, 10]
```

3 Dictionary

3.0.1 Q1: Key with maximum unique values

Given a dictionary with values list, extract key whose value has most unique values.

Example 1:

Input:

```
test_dict = {"CampusX" : [5, 7, 9, 4, 0], "is" : [6, 7, 4, 3, 3], "Best" : [9, 9, 6, 5, 5]}
```

Output:

```
CampusX
```

Example 2:

Input:

```
test_dict = {"CampusX" : [5, 7, 7, 7, 7], "is" : [6, 7, 7, 7], "Best" : [9, 9, 6, 5, 5]}
```

Output:

```
Best
```

[ ]:

```python
# write your code here
test_dict = {"CampusX" : [5, 7, 7, 7, 7], "is" : [6, 7, 7, 7], "Best" : [9, 9, 6, 5, 5]}

max_val = 0
max_key = ''
for i in test_dict:
  if max_val < len(set(test_dict[i])):
    max_val = len(set(test_dict[i]))
    max_key = i

print(max_key)
```

```
Best
```

3.0.2 Q2: Replace words from Dictionary. Given String, replace it's words from lookup dictionary.

Example 1:

Input:

```
test_str = 'CampusX best for DS students.'
repl_dict = {"best" : "is the best channel", "DS" : "Data-Science"}
```

Output:

```
CampusX is the best channel for Data-Science students.
```

Example 2:

Input:

```
test_str = 'CampusX best for DS students.'
repl_dict = {"good" : "is the best channel", "ds" : "Data-Science"}
```

Output:

```
CampusX best for DS students.
```

[ ]:

```python
# write your code here
test_str = 'CampusX best for DS students.'
repl_dict = {"best" : "is the best channel", "DS" : "Data-Science"}

res = []
for i in test_str.split():
  if i in repl_dict:
    res.append(repl_dict[i])
  else:
    res.append(i)
```

```
print(" ".join(res))
```

```
CampusX is the best channel for Data-Science students.
```

3.0.3 Q3: Convert List to List of dictionaries. Given list values and keys list, convert these values to key value pairs in form of list of dictionaries.

Example 1:

Input:

```
test_list = ["DataScience", 3, "is", 8]
key_list = ["name", "id"]
```

Output:

```
[{'name': 'DataScience', 'id': 3}, {'name': 'is', 'id': 8}]
```

Example 2:

Input:

```
test_list = ["CampusX", 10]
key_list = ["name", "id"]
```

Output:

```
[{'name': 'CampusX', 'id': 10}]
```

[ ]:

```
# write your code here
test_list = ["CampusX", 10]
key_list = ["name", "id"]

n = len(test_list)

res = []

for i in range(0,n,2):
  res.append({key_list[0]: test_list[i],key_list[1]:test_list[i+1]})

print(res)
```

```
[{'name': 'CampusX', 'id': 10}]
```

3.0.4 Q4: Convert a list of Tuples into Dictionary.

Example 1:

Input:

```
[("akash", 10), ("gaurav", 12), ("anand", 14), ("suraj", 20), ("akhil", 25), ("ashish", 30)]
```

Output:

```
{'akash': [10], 'gaurav': [12], 'anand': [14], 'suraj': [20], 'akhil': [25], 'ashish': [30]}
```

Example 2:

Input:

`[('A', 1), ('B', 2), ('C', 3)]`

Output:

`{'A': [1], 'B': [2], 'C': [3]}`

[ ]:

```python
# write your code here
L1 = [("akash", 10), ("gaurav", 12), ("anand", 14), ("suraj", 20), ("akhil",␣
 ↪25), ("ashish", 30)]
L = [('A', 1), ('B', 2), ('C', 3)]
d = {}
for i,j in L:
  d[i] = [j]

print(d)
```

```
{'A': [1], 'B': [2], 'C': [3]}
```

3.0.5 Q5: Sort Dictionary key and values List.

Example 1:

Input:

`{'c': [3], 'b': [12, 10], 'a': [19, 4]}`

Output:

`{'a': [4, 19], 'b': [10, 12], 'c': [3]}`

Example 2:

Input:

`{'c': [10, 34, 3]}`

Output:

`{'c': [3, 10, 34]}`

[ ]:

```python
# write your code here

d = {'c': [3], 'b': [12, 10], 'a': [19, 4]}

res = {}

for i in sorted(d):
  res[i] = sorted(d[i])

print(res)
```

```
{'a': [4, 19], 'b': [10, 12], 'c': [3]}
```

[ ]:

```

```

session-6-task-solutions

February 13, 2024

0.0.1 `Problem-1`: Write a Python function that takes a list and returns a new list with unique elements of the first list.

Exercise 1:

Input:

`[1,2,3,3,3,3,4,5]`

Output:

`[1, 2, 3, 4, 5]`

[ ]:
```
# Write code here
def return_unique(L):
  res = []

  for i in L:
    if i not in res:
      res.append(i)

  return res

L = [1,2,3,3,3,3,4,5]
return_unique(L)
```

[ ]:
```
[1, 2, 3, 4, 5]
```

0.0.2 `Problem-2`: Write a Python function that accepts a hyphen-separated sequence of words as parameter and returns the words in a hyphen-separated sequence after sorting them alphabetically.

Example 1:

Input:

`green-red-yellow-black-white`

Output:

`black-green-red-white-yellow`

[ ]:

```python
# Write code here

def sort_sequence(seq):
  temp = []

  for i in sorted(seq.split('-')):
    temp.append(i)

  return "-".join(temp)

s = 'green-red-yellow-black-white'
sort_sequence(s)
```

[ ]:

```
'black-green-red-white-yellow'
```

0.0.3 Problem 3: Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

```
Sample String : 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
Expected Output :
No. of Upper case characters :  9
No. of Lower case Characters :  47
```

[ ]:

```python
# Write code here

def lower_upper(s):
  lower_count = 0
  upper_count = 0

  for i in s:
    if i.islower():
      lower_count += 1
    elif i.isupper():
      upper_count += 1
    else:
      pass

  return lower_count,upper_count

s = 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
x,y = lower_upper(s)
print('No. of Lower case characters:', x)
print('No. of Upper case Characters:', y)
```

```
No. of Lower case characters: 47
No. of Upper case Characters: 9
```

0.0.4 Problem 4: Write a Python program to print the even numbers from a given list.

```
Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9]
Expected Result : [2, 4, 6, 8]
```

[ ]:

```
# Write code here

def is_even(L):
  res = []

  for i in L:
    if i % 2 == 0:
      res.append(i)

  return res

is_even([1,2,3,4,5,6,7])
```

[ ]:

```
[2, 4, 6]
```

0.0.5 Problem 5: Write a Python function to check whether a number is perfect or not.

A Perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example :

```
The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2
Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 +

The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496
```

[ ]:

```
# Write code here
def perfect_num(num):

  sum = 0

  for i in range(1,num):
    if num % i == 0:
      sum += i

  return sum == num

perfect_num(29)
```

[ ]:

```
False
```

0.0.6 Problem-6: Write a Python function to concatenate any no of dictionaries to create a new one.

```
Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

[ ]:

```
# Write code here

def merge_dict(*kwargs):
  d = {}

  for i in kwargs:
    d.update(i)

  return d



dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}

merge_dict(dic1,dic2,dic3)
```

[ ]:

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Problem-7 Write a python function that accepts a string as input and returns the word with most occurence.

```
Input:
hello how are you i am fine thank you
```

```
Output
you -> 2
```

[ ]:

```
# Write code here
def most_used(s):

  d = {}
  for i in s.split():
    if i in d:
      d[i] = d[i] + 1
    else:
      d[i] = 1

  max_val = max(d.values())
  for i in d:
    if d[i] == max_val:
      print(i,'->',d[i])
      break

most_used('hello hello hello you i am fine thank you')
```

```
hello -> 3
```

`Problem-8` Write a python function that receives a list of integers and prints out a histogram of bin size 10

```
Input:
[13,42,15,37,22,39,41,50]
```

```
Output:
{11-20:2,21-30:1,31-40:2,41-50:3}
```

[ ]:

```
# Write code here
import math

def histogram(L):

  min_bin = math.floor(min(L)/10)*10
  max_bin = math.ceil(max(L)/10)*10

  d={}

  for i in range(min_bin,max_bin,10):
    count = 0
    for j in L:
      if i+1<=j<=i+10:
        count+=1

    d[str(i+1) + '-' + str(i+10)] = count

  return d

histogram([13,42,15,37,22,39,41,50])
```

```
[ ]: {'11-20': 2, '21-30': 1, '31-40': 2, '41-50': 3}
```

`Problem-9` Write a python function that accepts a list of 2D co-ordinates and a query point, and then finds the the co-ordinate which is closest in terms of distance from the query point.

```
List of Coordinates
[(1,1),(2,2),(3,3),(4,4)]
Query Point
(0,0)
```

```
Output
Nearest to (0,0) is (1,1)
```

[ ]:

```
# Write code here

def shortest_dist(points,query):
  temp = []
  for i in points:
    distance = ((i[0] - query[0])**2 + (i[1] - query[1])**2)**0.5
    temp.append(distance)

  return points[sorted(list(enumerate(temp)),key=lambda x:x[1])[0][0]]


points = [(1,4),(2,-2),(13,3),(14,4)]
query = (0,0)

shortest_dist(points,query)
```

[ ]:

```
(2, -2)
```

`Problem-10`:Write a python program that receives a list of strings and performs bag of word operation on those strings

https://en.wikipedia.org/wiki/Bag-of-words_model

[ ]:

```
# Write code here

def bow(L):

  vocab = set()

  for i in L:
    vocab.update(i.split())

  result = []

  for i in L:
    result.append([])
    for j in vocab:
      result[-1].append(i.count(j))

  print(vocab)
  return result


L = [
    'cat mat rat cat',
     'sat bat fat cat rat',
     'pat cat mat rat'
]

bow(L)
```

```
{'bat', 'sat', 'mat', 'cat', 'pat', 'rat', 'fat'}
```

[ ]:
```
[[0, 0, 1, 2, 0, 1, 0], [1, 1, 0, 1, 0, 1, 1], [0, 0, 1, 1, 1, 1, 0]]
```

###Problem 11: Write a Python program to add three given lists using Python map and lambda.

[ ]:
```
# Write code here
L1 = [1,2,3]
L2 = [4,5,6]
L3 = [7,8,9]

# [12,15,18]

list(map(lambda x,y,z:x+y+z,L1,L2,L3))
```

[ ]:
```
[12, 15, 18]
```

###Problem-12:Write a Python program to create a list containing the power of said number in bases raised to the corresponding number in the index using Python map. Input:

list1 = [1,2,3,4,5,6]

Output:

[1,2,9,64,625,-]

[ ]:
```
# Write code here
list1 = [1,2,3,4,5,6]
list(map(lambda x,y:x**y,list1,range(len(list1))))
```

[ ]:
```
[1, 2, 9, 64, 625, 7776]
```

###Problem-13 Using filter() and list() functions and .lower() method filter all the vowels in a given string.

[ ]:
```
# Write code here
str1="FIFA world cup in 2022 will take place in Qatar"

list(filter(lambda x:True if x.lower() in 'aeiou' else False,str1))
```

[ ]:
```
['I', 'A', 'o', 'u', 'i', 'i', 'a', 'e', 'a', 'e', 'i', 'a', 'a']
```

Problem-14: Use reduce to convert a 2D list to 1D

[ ]:

```
# Write code here
ini_list = [[1, 2, 3],
            [3, 6, 7],
            [7, 5, 4]]

import functools
functools.reduce(lambda x,y:x+y,ini_list)
```

[ ]:

```
[1, 2, 3, 3, 6, 7, 7, 5, 4]
```

`Problem 15-` A dictionary contains following information about 5 employees: - First name - Last name - Age - Grade(Skilled,Semi-skilled,Highly skilled) Write a program using map/filter/reduce to a list of employees(first name + last name) who are highly skilled

[ ]:

```
# Write code here
employees = [
    {
        'fname':'Nitish',
        'lname':'Singh',
        'age' : 33,
        'grade':'skilled'
    },
    {
        'fname':'Ankit',
        'lname':'Verma',
        'age' : 34,
        'grade':'semi-skilled'
    },
    {
        'fname':'Neha',
        'lname':'Singh',
        'age' : 35,
        'grade':'highly-skilled'
    },
    {
        'fname':'Anurag',
        'lname':'Kumar',
        'age' : 30,
        'grade':'skilled'
    },
    {
        'fname':'Abhinav',
        'lname':'Sharma',
        'age' : 37,
        'grade':'highly-skilled'
    }
]
```

[ ]:

```
list(map(lambda x:x['fname'] + ' ' + x['lname'],list(filter(lambda x:True if␣
  ↪x['grade'] == 'highly-skilled' else False,employees))))
```

[ ]:

```
['Neha Singh', 'Abhinav Sharma']
```

[ ]:

```

```