WEEK 1

Write a python program to print two strings/objects (“This is our” , “Second Program”) by separator “,” and supress the newline character and print “ENDLINE” at the end

```
print("This is our","First Program\nENDLINE",sep = ",")
```

```
This is our,First Program
ENDLINE
```

WEEK 2

Python Program to Check if a Number is Odd or Even

```
n = int(input("enter a number: "))
if n%2 == 0:
  print(n,"is a even number")
else:
  print(n,"is a odd number")
```

```
enter a number: 4
4 is a even number
```

Python program to solve quadratic equation

```
i = int(input("first value: "))
j = int(input("second value: "))
k = int(input("third value: "))
d = (j**2) - (4*i*k)
r1 = (-j + (d**0.5))/(2*i)
r2 = (-j - (d**0.5))/(2*i)
if d > 0:
  print("real roots: ",r1,r2)
elif d < 0:
  print("complex roots: ",r1,r2)
else:
  print("roots are real and equal: ",(-j/(2*i)),(-j/(2*i)))
```

```
first value: 1
second value: -2
third value: 1
roots are real and equal:  1.0 1.0
```

Python program to find the area of a triangle

```
p = float(input("enter first integer value: "))
q = float(input("enter second integer value: "))
r = float(input("enter third integer value: "))
```

```
r = float(input(" enter third integer value:  "))
s = (p+q+r)/2
area = (s*(s-p)*(s-q)*(s-r))**0.5
print("area of the triangle is: ",area)
```

```
enter first integer value: 3
enter second integer value: 4
enter third integer value: 5
area of the triangle is:  6.0
```

Python program to swap two variables.

```
a1 = int(input("enter first integer value: "))
a2 = int(input("enter second integer value: "))
print("values before swapped are: ",a1, a2)
temp = a1
a1 = a2
a2 = temp
print("values after swapped are: ",a1, a2)
```

```
enter first integer value: 10
enter second integer value: 99
values before swapped are:  10 99
values after swapped are:  99 10
```

Python program to convert kilometres to miles

```
n = float(input("enter the kilometers: "))
print("kilometers to miles: ",n*0.621)
```

```
enter the kilometers: 5
kilometers to miles:  3.105
```

Python Program to Check Leap Year

```
year = int(input("enter a year: "))
if year%4 == 0:
  print("{} is a leap year.".format(year))
elif year%100 == 0:
  print("{} is not a leap year.".format(year))
elif year%400 == 0:
  print("{} is a leap year.".format(year))
else:
  print("{} is not a leap year.".format(year))
```

```
enter a year: 2028
2028 is a leap year.
```

Python Program to Check Armstrong Number

```
import math
```

```
p = int(input("enter a number: ")); x = p; sum = 0
while p!=0:
  last = p%10
  sum = sum + math.pow(last,3)
  p = p//10
if sum == x:
  print(x,"is an armstrong number")
else:
  print(x,"is not an armstrong number")
```

```
enter a number: 153
153 is an armstrong number
```

Python Program to Check if a Number is Positive, Negative or Zero

```
n = int(input("enter a number: "))
if n > 0:
  print("{} is a positive number.".format(n))
elif n < 0:
  print("{} is a negative number.".format(n))
elif n == 0:
  print("it's a zero.")
else:
  print(" input error ")
```

```
enter a number: 10
10 is a positive number.
```

WEEK 3

Accept the following from the user and calculate the percentage of class attended: a. Total number of working days b. Total number of days for absent After calculating percentage show that, If the percentage is less than 75, than student will not be able to sit in exam.

```
t = int(input("enter total no.of working days: "))
a = int(input("enter total no.of days for absent: "))
per = ((t-a)/t)*100
if per < 75:
  print("the student will not be able to sit in exam hall.")
else:
  print("the student is able sit in the exam hall")
```

```
enter total no.of working days: 28
enter total no.of days for absent: 5
the student is able sit in the exam hall
```

Write a program to accept the cost price of a bike and display the road tax to be paid according to the .

```
n = float(input("enter the cost price of a bike: "))
if n > 100000:
  print("road tax to be paid: ",n+(n*0.15))
elif n > 50000 and n <= 100000:
  print("road tax to be paid: ",n+(n*0.1))
elif n <= 50000:
  print("road tax to be paid: ",n+(n*0.05))
else:
  print(" input error or data wrong")
```

```
enter the cost price of a bike: 27000
road tax to be paid:  28350.0
```

**Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range print an error. If the score is between 0.0 and 1.0, print a grade using the following Score Grade: Score Grade

> = 0.9 A, >= 0.8 B, >= 0.7 C, >= 0.6 D, < 0.6 F**

```
n = float(input("enter the score: "))
if n >= 0.0 and n <= 1.0:
  if n >= 0.9:
    print("A GRADE")
  elif n >= 0.8:
    print("B GRADE")
  elif n >= 0.7:
    print("C GRADE")
  elif n >= 0.6:
    print("D GRADE")
  elif n < 0.6:
    print("F GRADE")
  else:
    print(" error ")
else:
  print(" ERROR ")
```

```
enter the score: 1.0
A GRADE
```

Write an interactive program to read an integer. If it is positive then display the corresponding binary representation of that number. The user must enter 999 to stop. In case the user enters a negative number, then ignore that input and ask the user to renter any different number

```
while True:
  p = int(input("enter a number: "))
  if p < 0:
    continue
  if p == 999:
    break
  print("binary number: ",bin(p))
```

```
enter a number: -1
enter a number: -1
enter a number: 15
binary number:  0b1111
enter a number: 16
binary number:  0b10000
enter a number: 999
```

The ExConFair is the region's largest trade fair on Construction Equipment & Technology. The Event organizers hired college students as volunteers to work at the fair as the event is targeted to be attended by approx. 30 million visitors. At the Office in the fair, there are two guards who count how many times a volunteer enters into the fair ground. Though the duty of a guard is 24 hour in a day, but sometimes they fall asleep during their duty and could not track the entry of volunteers in the fair ground. But one better thing is that they never fall asleep at the same time. At least one of them remains awake and counts who enters into the office. Now the Event Head wants to calculate how many times a volunteer has entered into the fair ground. He asked to the guard and they give him two integers A and B, count of first guard and second guard respectively. Help the Event Head to count the minimum and maximum number of times a volunteer could have entered into the fair ground.

```
a = int(input("enter count of person A: "))
b = int(input("enter count of person B: "))
if a > b:
  print("minimum number of times is: ",a)
else:
  print("minimum number of times is: ",b)

print("maximum number of times is: ",a+b)
```

```
enter count of person A: 30
enter count of person B: 40
minimum number of times is:  40
maximum number of times is:  70
```

WEEK 4

Write a python program to construct the following patterns using nested for loop

pattern 1

```
for i in range(1,6):
  for j in range(1,i+1):
    print("*",end = " ")
  print()
for k in range(1,6):
  for l in range(6,k,-1):
    print("*",end = " ")
```

```
  print()
```

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

pattern 2

```
for i in range(1,7):
  for j in range(1,i+1):
    print(j,end = " ")
  print()
```

```
     1
     1 2
     1 2 3
     1 2 3 4
     1 2 3 4 5
     1 2 3 4 5 6
```

pattern 3

```
for i in range(1,7):
  for j in range(1,i+1):
    print(i,end = " ")
  print()
```

```
     1
     2 2
     3 3 3
     4 4 4 4
     5 5 5 5 5
     6 6 6 6 6 6
```

Write a python program to print Multiplication table using format function inside print

```
n = int(input("enter a number: "))
print("multiplication table of {} is".format(n))
for i in range(1,11):
  print("{} x {} = {}".format(n,i,n*i))
```

```
     enter a number: 2
     multiplication table of 2 is
     2 x 1 = 2
     2 x 2 = 4
     2 x 3 = 6
```

```
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

Write a Python Program to Check how many times a given number can be divided by 3 before it is less than or equal to 10.

```
n = int(input("enter a number: "))
count = 0
while n >= 10:
  n = n//3
  count = count + 1
print("no.of times a given number can be divided by 3: ",count)
```

```
enter a number: 234
no.of times a given number can be divided by 3:  3
```

Calculate the sum of the first 20 items of 1 + 2 + 4 + 8 + 16 +... using while True infinite loop with break statement.

```
n = int(input("enter how many times: "))
i = 0; sum = 0; count = 0
while True:
  sum = sum + 2**i
  count = count + 1
  if count == n:
    break
  i = i + 1

print("sum of first {} items: {}".format(n,sum))
```

```
enter how many times: 20
sum of first 20 items: 1048575
```

Transform the existing while loops with calculations 0-100, and add continue statements to calculate only the odd sums.

```
odd = 0
for i in range(0,101):
  if i%2 == 0:
    continue
  else:
    odd = odd + i
print("sum of all odd's from 0 to 100: ",odd)
```

```
sum of all odd's from 0 to 100:  2500
```

WEEK 5 AND WEEK 6

Understand the usage of methods in python a. Create a Python script with two methods which will compute the area and the perimeter of a circle. b. Create a Python script which takes length & width from the user and two methods which will compute the area & perimeter of a rectangle

```
# method 1
def circle(r1):
  area = 3.14*r1*r1
  perimeter = 2*3.14*r1
  print("area and perimeter of the circle: ",round(area,2),perimeter,sep = ",")
r = float(input("enter the radius of the circle: "))
circle(r)
```

```
enter the radius of the circle: 3
area and perimeter of the circle: ,28.26,18.84
```

```
# method 2
r = float(input("enter the radius of the circle: "))
a = 3.14*r*r
p = 2*3.14*r
print("area and perimeter of the circle: ",a,p,sep = ",")
```

```
enter the radius of the circle: 3
area and perimeter of the circle: ,28.259999999999998,18.84
```

```
# method 1
def rect(a,b):
  area = a*b
  perimeter = 2*(a+b)
  print("area and perimeter of the rectangle: ",area,perimeter,sep = ",")
l = float(input("enter length of the rectangle: "))
b = float(input("enter bredth of the rectangle: "))
rect(l,b)
```

```
enter length of the rectangle: 3
enter bredth of the rectangle: 5
area and perimeter of the rectangle: ,15.0,16.0
```

```
# method 2
l = float(input("enter length of the rectangle: "))
b = float(input("enter bredth of the rectangle: "))
a = l*b
p = 2*(l+b)
print("area and perimeter of the rectangle: ",a,p,sep = ",")
```

```
enter length of the rectangle: 3
enter bredth of the rectangle: 5
area and perimeter of the rectangle: ,15.0,16.0
```

Write recursive functions in python to compute (i) Greatest Common Divisior GCD of two numbers (ii) Least Common Multiple of two numbers (iii) Factorial of a number (iv) Fibonacci series

1. recursive function for GCD

```
def gcd(a,b):
  if b == 0:
    return a
  else:
    return gcd(b,a%b)
i = int(input("enter first number: "))
j = int(input("enter second number: "))
print("GCD of two numbers: ",gcd(i,j))
```

```
enter first number: 2
enter second number: 4
GCD of two numbers:  2
```

2. recursive function for LCM

```
def lcm(x,y):
  if x > y:
    min = x
  else:
    min = y
  while 1:
    if min%x == 0 and min%y == 0:
      print("LCM of two numbers: ",min)
      break
    min += 1
a = int(input("enter first number: "))
b = int(input("enter second number: "))
lcm(a,b)
```

```
enter first number: 3
enter second number: 4
LCM of two numbers:  12
```

3. recursive function for factorial

```
def fact(x):
  if x == 0:
    return 1
  else:
    return x*fact(x-1)
n = int(input("enter a number: "))
print("factorial of the number: ",fact(n))
```

```
enter a number: 5
factorial of the number:  120
```

4. recursive function for fibonacci series

```
def fib(n):
  if n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    return (fib(n-1) + fib(n-2))
x = int(input("enter a number: "))
print("fibonacci series upto",x,": ")
for i in range(x):
  print(fib(i),end = ",")
```

```
enter a number: 10
fibonacci series upto 10 :
0,1,1,2,3,5,8,13,21,34,
```

Solve the following tasks by using different types of using arguments in python.

a. Create a function showEmployee() in such a way that it should accept employee name, and its salary and display both. If the salary is missing in the function call assign default value 10,000 to salary. Hint: Use the concept of default length argument concept in python.

```
def showEmployee(x,y = 10000):
  print("name of the employee: ",x)
  print("salary of a employee: ",y)

a = input("enter the employee name: ")
b = float(input("enter the employee salary: "))
showEmployee(a,b)
```

```
enter the employee name: mounav
enter the employee salary: 25000
name of the employee:  mounav
salary of a employee:  25000.0
```

b. Design a python function to read individual subject marks of a B.Tech student. Calculate total marks and average marks. The number of subjects may vary over the semester. You need to design a function so that function can be applicable for any number of subjects. Hint: Use the concept of Variable length argument concept in python

```
def student(*marks):
  print("marks of students are: ",marks)
  sum = 0
  num = 0
  for i in marks:
```

```
  for i in marks:
    num += 1
    sum += i
  avg = sum/num
  print("the average of marks: ",avg)

student(95,90,99,85,89,100)
```

```
marks of students are:  (95, 90, 99, 85, 89, 100)
the average of marks:  93.0
```

You are given a date. Your task is to find what the day is on that date. Input Format A single line of input containing the space separated month, day and year, respectively, in DD MM YY format. Constraints : 2000<year<3000

Sample Input 08 05 2015

Sample Output FRIDAY

```
import datetime
from datetime import date
date=str(input('Enter the date(for example:09 02 2019):'))

day, month, year = date.split(' ')

day_name = datetime.date(int(year), int(month), int(day))

print("day of the date: ",day_name.strftime("%A"))
```

```
Enter the date(for example:09 02 2019):08 05 2015
day of the date:  Friday
```

You are given two integer arrays, and of dimensions X Your task is to perform the following operations: Add (A+B) Subtract (A – B) Multiply (A * B) Integer Division (A / B) Mod (A% B) Power (A**B)

Note: There is a method numpy.floor_divide() that works like numpy.divide() except it performs a floor division.

Input Format

The first line contains two space separated integers, and. the next lines contains space separated integers of array . The following lines contains space separated integers of array

Output Format

Print the result of each operation in the given order under Task.

Sample Input

1 4

1 2 3 4

5 6 7 8

Sample Output

[[6 8 10 12]]

[[-4 -4 -4 -4]]

[[5 12 21 32]]

[[0 0 0 0]]

[[1 2 3 4]]

[[1 64 2187 65536]]

```
import numpy
n,m = map(int,input("enter two numbers seperated by space: ").split())

a = numpy.array([input("enter array 1 numbers: ").split() for i in range(n)],int)

b = numpy.array([input("enter array 2 numbers: ").split() for j in range(n)],int)

print("array 1 + array 2 = ",a + b)
print("array 1 - array 2 = ",a - b)
print("array 1 * array 2 = ",a * b)
print("array 1 // array 2 = ",a // b)
print("array 1 % array 2 = ",a % b)
print("array 1 ** array 2 = ",a ** b)
```

```
enter two numbers seperated by space: 1 4
enter array 1 numbers: 1 2 3 4
enter array 2 numbers: 5 6 7 8
array 1 + array 2 =  [[ 6  8 10 12]]
array 1 - array 2 =  [[-4 -4 -4 -4]]
array 1 * array 2 =  [[ 5 12 21 32]]
array 1 // array 2 =  [[0 0 0 0]]
array 1 % array 2 =  [[1 2 3 4]]
array 1 ** array 2 =  [[    1    64  2187 65536]]
```

WEEK 7

Write a python program to calculate the class average by taking students name as string, student roll number as integer, five subject marks from the user and display name, roll number, marks, all subjects marks and average with precision with 2 using string formatting.

(Hint: String format %s, %d, %f)

```
name = input("enter the student name: ")
roll = int(input("enter the student roll number: "))
marks = []
for i in range(1,6):
  a = float(input('enter the marks of subject {}: '.format(i)))
  marks.append(a)
print("student name : %s"%(name))
print("student roll number : %d"%(roll))
```

```
print("student roll number : %d"%(roll))
print("student 5 subject marks are: ",marks)
avg = sum(marks)/5
print("average of 5 subjects is: %.2f"%(avg))
```

```
enter the student name: mounav
enter the student roll number: 22
enter the marks of subject 1: 100
enter the marks of subject 2: 98
enter the marks of subject 3: 97
enter the marks of subject 4: 99
enter the marks of subject 5: 93
student name : mounav
student roll number : 22
student 5 subject marks are:  [100.0, 98.0, 97.0, 99.0, 93.0]
average of 5 subjects is: 97.40
```

Given a string and two substrings, write a Python program to extract the string between the found two substrings

```
string = input("enter any english statement: ")
str1 = "is"
str2 = "and"
idx1 = string.index(str1)
idx2 = string.index(str2)
print("extracted string is: ",string[idx1+len(str1)+1:idx2])
```

```
enter any english statement: vr is best and autonamous
extracted string is:  best
```

Given Strings with words, the task is to write a Python program to split each word into two halves on the basis of assigned percentages according to the given values. (Sample Input: "VR Siddhartha Engineering College is Best College in AP", per=50 Output: " V R Siddh artha Engine eering Coll ege i s Be st Coll ege i n A P")

```
n = input("enter any english statement: ").split()
#print(n)
percent = int(input("enter the percentage for string: "))

result = ""
for i in n:
  p = int((percent/100)*len(i))
  s1 = i[:p]
  s2 = i[p:]

  result = result + s1 + " " + s2 + " "

print("each word splitted: ",result)
```

```
enter any english statement: VR Siddhartha Engineering College is Best College in AP
enter the percentage for string: 50
each word splitted:  V R Siddh artha Engin eering Col lege i s Be st Col lege i n A P
```

Write a Python program to find all adverbs and their positions in a given sentence. (Sample Input: "Clearly, he has no excuse for such behavior." Output: 0-7: Clearly)

```
import re
n = input("write any english statement: ")

for m in re.finditer(r"\w+ly",n):
  print("adverbs are: %d-%d:%s"%(m.start(),m.end(),m.group(0)))
```

```
write any english statement: Clearly, he has no excuse for such behavior
adverbs are: 0-7:Clearly
```

Given a string, the task is to write a Python program to extract date from it. Sample Input: "VRSEC at 2021-07-06" Output: 0-7: 2021-07-06, Hint: re.search(), strptime())

```
import re
from datetime import datetime

n = input("write any statement: ")
match = re.search(r'\d{4}-\d{2}-\d{2}',n)

date = datetime.strptime(match.group(),'%Y-%m-%d').date()
print("extracted date is: ",date)
```

```
write any statement: VRSEC at 2021-07-06
extracted date is:  2021-07-06
```

WEEK 8

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34, 67, 55, 33, 12, 98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34',67', '55', '33', '12', '98').

```
w = input("enter a group of numbers:  ").split(",")
#print(w)

s1 = list(w)
s2 = tuple(w)

print("list elements are: ",s1)
print("tuple elements are: ",s2)
```

```
enter a group of numbers:  34,67,55,33,12,98
['34', '67', '55', '33', '12', '98']
list elements are:  ['34', '67', '55', '33', '12', '98']
tuple elements are:  ('34', '67', '55', '33', '12', '98')
```

Write a python program to find tuples which have all elements divisible by K from a list of tuples.

```
n = [(12,45,63),(3,5,9),(15,27,61),(90,72,24)]
k = int(input("enter a number to check divisibility: "))

for i in n:
  c = 0
  for j in i:
    if j%k!=0:
      c = 1
      break

  if c == 0:
    print(i,end=" ")
```

```
 enter a number to check divisibility: 3
 (12, 45, 63) (90, 72, 24)
```

Write a python program to find Tuples with positive elements in List of tuples.

```
n = [(2,-1,3),(3,5,9),(5,7,1),(-1,-2,4),(0,3,6)]

for i in n:
  c = 0
  for j in i:
    if j <= 0:
      c = 1
      break

  if c == 0:
    print(i,end=" ")
```

```
 (3, 5, 9) (5, 7, 1)
```

Write a python program remove duplicate tuples from list of tuples.

```
n = [(1, 2), (5, 7), (3, 6), (1,2),(5,7)]
s1 = set(n)
s2 = list(s1)
'''
for i in range(0,len(n)):
    if n[i] == n[i+1]:
        del n[i]
    else:
        print(n[i])    '''

print("list of tuples without duplicates: ",s2)
```

```
 list of tuples without duplicates:  [(1, 2), (5, 7), (3, 6)]
```

Python program in which we need to perform Row-wise custom elements addition in Tuple matrix

```
hd = [ [('abc',1), ('def',2)], [('pqr',9)], [('xyz',5),('klm',1)] ]
print("the original list: ",hd)

n = [1,8,0]

#w = zip(hd,n)
#w = list(w)
#print()
#print(w)

print()
ans = [ [(idx,val) for idx in key] for key, val in zip(hd,n)]
print("the matrix after row elements addition: ",ans)
```

```
the original list:  [[('abc', 1), ('def', 2)], [('pqr', 9)], [('xyz', 5), ('klm', 1)]]

the matrix after row elements addition:  [[(('abc', 1), 1), (('def', 2), 1)], [(('pqr
```

Python program to extract digits from Tuple list (I/p: [(1,3), (4,5] | o/p: [1,3,4,5]

```
from itertools import chain

s = [(1,3),(4,5)]
print("original list: ",s)

c = list(chain.from_iterable(s))
print("extracted digits are: ",c)
```

```
original list:  [(1, 3), (4, 5)]
extracted digits are:  [1, 3, 4, 5]
```

```
from itertools import chain

s = [(1,3),(4,5)]
print("original list: ",s)

f = lambda x: str(x)
c = list(chain.from_iterable(s))

temp = list(map(f,c))
ans = set()

for i in temp:
  for j in i:
    ans.add(i)

print("extracted digits are: ",ans)
```

```
original list:  [(1, 3), (4, 5)]
extracted digits are:  {'3', '4', '5', '1'}
```

WEEK 9

Write a Python script to sort (ascending and descending) a dictionary by value.

```
import operator as op
d = {1:2,3:4,4:3,2:1,0:0}
print("original dictionary: ",d)
sd = sorted(d.items(),key=op.itemgetter(1))
print("ascending dictionary: ",sd)

sd1 = sorted(d.items(),key=op.itemgetter(1),reverse=1)
print("decending dictionary: ",sd1)
```

```
original dictionary:  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
ascending dictionary:  [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]
decending dictionary:  [(3, 4), (4, 3), (1, 2), (2, 1), (0, 0)]
```

Write a Python program to convert string values of a given dictionary, into integer/float datatypes.

```
def conv_int(lst):
  r = [ dict( [a,int(x)] for a,x in b.items() ) for b in lst ]
  return r

def conv_float(lst):
  r = [ dict( [a,float(x)] for a,x in b.items() ) for b in lst ]
  return r

nums = [{ 'x':'10' , 'y':'20' , 'z':'30' }, { 'p':'40', 'q':'50', 'r':'60'}]
print("original list: ",nums)

print("\ninteger values of dictionary: ",conv_int(nums))

print("\nfloat values of dictionary: ",conv_float(nums))
```

```
original list:  [{'x': '10', 'y': '20', 'z': '30'}, {'p': '40', 'q': '50', 'r': '60'}]

integer values of dictionary:  [{'x': 10, 'y': 20, 'z': 30}, {'p': 40, 'q': 50, 'r': 6

float values of dictionary:  [{'x': 10.0, 'y': 20.0, 'z': 30.0}, {'p': 40.0, 'q': 50.0
```

Given a list and dictionary, map each element of list with each item of dictionary, forming nested dictionary as value.

```
d = {'a':92,'b':93,'c':94}
```

```
l = ['A','B','C']
r = {}

for k,v in zip(l,d.items()):
  r[k] = dict([v])

print("the mapped dictionary: ",r)
```

```
the mapped dictionary:  {'A': {'a': 92}, 'B': {'b': 93}, 'C': {'c': 94}}
```

Given a list, write a Python program to convert the given list to dictionary such that all the odd elements have the key, and even number elements have the value. Since python dictionary is unordered, the output can be in any order.

```
l = [1,2,3,4,5,6,7,8,9,10]
print("given list is: ",l)
value = []
key = []
for i in l:
  if i%2 == 0:
    value.append(i)
  else:
    key.append(i)

#print("value: \n",value,"\nkey: \n",key)
s = zip(key,value)
s = dict(s)
print("the dictionary: ",s)
```

```
given list is:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
the dictionary:  {1: 2, 3: 4, 5: 6, 7: 8, 9: 10}
```

Create a dictionary from two lists of same length, considering the first list elements as keys and second list as values of new dictionary.

```
keys = ['vamsi','mounav','ram','bhargav','charan','ajay','rizwan']
values = [90,100,99,91,95,100,100]

d = zip(keys,values)
d = dict(d)
#res = {keys[i]: values[i] for i in range(len(keys))}

print("dictionary of student marks is: ",d)
```

```
dictionary is:  {'vamsi': 90, 'mounav': 100, 'ram': 99, 'bhargav': 91, 'charan': 95,
```

Given an input string and a pattern, check if characters in the input string follows the same order as determined by characters present in the pattern. Assume there won't be any duplicate characters in the pattern

```
def follow(s,t):
  if len(s) < len(t):
    return False

  for i in range(len(t)-1):

    x = t[i]
    y = t[i+1]

    right = s.index(x)
    left = s.index(y)

    if right == -1 or left == -1 or right > left:
      return False

  return True

s = 'hello world'
t = "hw"

t1 = follow(s,t)
print("the string follows the pattern: ",t1)
```

```
the string follows the pattern:  True
```

✓ 0s completed at 7:10 PM