

PYTHON PROGRAMMING LAB ASSIGNMENT - 1

BY,

ROLL NO: 20VV1A1263.

NAME : VENKATA SAI YASWANTH BATTU.

B.TECH - INFORMATION TECHNOLOGY.

1 – II.

QUESTION NO: 1

AIM: Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.

DESCRIPTION:

The program is developed in python language that takes input from the user of weight in kilograms and displays the output converted into pounds.

ALGORITHM:

STEP 1: START.

STEP 2: Declare variable that stores input given in kgs.

STEP 3: Type cast the input as by default it is string.

STEP 4: multiply the input with 2.2. and store it another variable.

STEP 5: Display Output.

STEP6: END.

PROGRAM:

```
wtkg = float(input("Input weight in kilograms :"))
```

```
wtp = wtkg*2.2
```

```
print("The weight in pounds is:",round(wtp,4))
```

OUTPUT:

```
: wtkg = float(input("Input weight in kilograms :"))  
wtp = wtkg*2.2  
print("The weight in pounds is:",round(wtp,4))
```

Input weight in kilograms :50

The weight in pounds is: 110.0

QUESTION NO: 2

AIM: Write a program that asks the user to enter three numbers (use three separate input

statements). Create variables called total and average that hold the sum and average of

the three numbers and print out the values of total and average.

DESCRIPTION:

The program that asks the user to enter three numbers into 3 different variables . using two variables that store the value of total an average.

ALGORITHM:

STEP 1: START.

STEP 2: Read x, y, z .

STEP 3: calculate $\text{total} = x + y + z$.

STEP 4: calculate $\text{average} = \text{total} / 3$

STEP 5: Print total and average.

STEP 6: END.

PROGRAM:

```
x = float(input("Enter a number :"))
```

```
y = float(input("Enter a number :"))
```

```
z = float(input("Enter a number :"))
```

```
total = (x + y + z)
```

average = total/3

print("The tota is :",total,"The average is
:",round(average,4))

OUTPUT:

```
x = float(input("Enter a number :"))
y = float(input("Enter a number :"))
z = float(input("Enter a number :"))

total = (x+y+z)
average = total/3

print("The tota is :",total,"The average is :",round(average,4))
```

```
Enter a number :1
Enter a number :4
Enter a number :3
The tota is : 8.0 The average is : 2.6667
```

QUESTION NO: 3

AIM: Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.

DESCRIPTION:

Program works by using a for loop and a range function starts iterating from 8 and goes upto 89 with a step of 3.

ALGORITHM:

STEP 1: START.

STEP 2: Using for loop and range function use a variable *i* that points to values in the range function.

STEP 3: for every i print the value of 'i' with end replaced by space to feel good.

STEP 4: END.

PROGRAM:

```
for i in range(8,90,+3):
```

```
    print(i,end=" ")
```

OUTPUT:

```
for i in range(8,90,+3):  
    print(i,end=" ")
```

```
8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80 83  
86 89
```

QUESTION NO: 4

AIM:) Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.

DESCRIPTION:

Program asks the user to input the user name and the number of the times the names to be printed.

ALGORITHM:

STEP 1: START.

STEP 2: Read name.

STEP 3: Read freq with type casting the input into int type.

STEP 4: initialise $i = 1$.

STEP 5: using a while loop with condition $i \leq \text{freq}$.

STEP 6: print name for every iteration.

STEP 7: $i = i + 1$ in every iteration.

STEP 8: END.

PROGRAM:

```
name = input("Input the name :")
```

```
freq = int(input("Input the number of times  
to print : "))
```

```
i = 1
```

```
while(i <= freq):
```

```
    print(name)
```

```
    i = i + 1
```

OUTPUT:

```
: name = input("Input the name :")
freq = int(input("Input the number of times to print : "))
i = 1
while(i<=freq):
    print(name)
    i = i+1
.
```

Input the name :YASH

Input the number of times to print : 4

YASH

YASH

YASH

YASH

QUESTION NO: 5

AIM: Use a for loop to print a triangle like the one below. Allow the user to specify how

high the triangle should be.

*

**

DESCRIPTION:

Program uses looping to perform a printing of specified pattern.

ALGORITHM:

STEP 1: START.

STEP 2: read height of int type.

STEP 3: using a for loop starts from $l = 1$ and upto height using a range method.

STEP 4: nest another for loop starting from l to i

STEP 5: print "*" with end separation with space " ".

STEP6: Print a new line '\n' after every complete iterations of nested loop.

STEP 7: END.

PROGRAM:

```
height = int(input("Enter the height of the  
triangle :"))
```

```
for i in range(1,height+1):
```

```
for j in range(1,i+1):
```

```
    print("*",end=" ")
```

```
print("\n")
```

OUTPUT:

```
height = int(input("Enter the height of the traingle :"))
for i in range(1,height+1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
```

Enter the height of the traingle :4

*

.

* *

* * *

* * * *

QUESTION NO: 6

AIM: Generate a random number between 1 and 10. Ask the user to guess the number and

print a message based on whether they get it right or not.

DESCRIPTION:

Program takes input for a number in range of 1 to 10 and responds according if matched or not matched with the program generated number.

ALGORITHM:

STEP 1: START.

STEP 2: import random module.

STEP 3: store a random number generated by randint method of random stored in variable random_no.

STEP 4: guess_no a variable that stores the int type input given by user as a guess number.

STEP 5: If guess_np == random_no

Print Correct.

STEP6: else

Print Wrong Guess and try again.

STEP 7: END.

PROGRAM:

```
import random
```

```
random_no = random.randint(1,10)
```

```
guess_no = int(input("Please input ur gess !  
:"))
```

```
if guess_no==random_no:
```

```
    print("Hurray ! you guessed it right .")
```

```
else:
```

```
    print("The random number is  
:",random_no)
```

```
    print("Try again !")
```

OUTPUT:

```
import random  
random_no = random.randint(1,10)  
guess_no = int(input("Please input ur gess ! :"))  
if guess_no==random_no:  
    print("Hurray ! you guessed it right .")  
else:  
    print("The random number is :",random_no)  
    print("Try again !")
```

```
Please input ur gess ! :8  
The random number is : 5  
Try again !
```

QUESTION NO: 7

AIM: Write a program that asks the user for two numbers and prints Close if the numbers

are within .001 of each other and Not close otherwise.

DESCRIPTION:

The program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise.

ALGORITHM:

STEP 1: START.

STEP 2: read a, b

STEP 3: if $a > b$

$\text{diff} = \text{round}(a-b, 4)$ which does calculation of finding difference between a and b and round off upto 4 decimal places.

STEP 4: else:

$\text{diff} = \text{round}(b-a, 4)$

operates same as in step 3, instead finding difference varies.

STEP 5: a if statement is used to check the diff calculated is ≤ 0.001 .

STEP 6: if yes, print close.

STEP 7: if no, print not close.

STEP 8: END.

PROGRAM:

```
a = float(input("enter a number"))
```

```
b = float(input("enter another number"))
```

```
if a > b:
```

```
    diff = round(a-b,4)
```

```
else:
```

```
    diff = round(b-a,4)
```

```
if diff <= 0.001:
```

```
    print("Close")
```

```
else:
```

```
    print("Not Close").
```

OUTPUT:

```
: a = float(input("enter a number"))
  b = float(input("enter another number"))
  if a>b:
      diff = round(a-b,4)
  else:
      diff = round(b-a,4)
  if abs(diff)<=0.001:
      print("Close")
  else:
      print("Not Close")
```

```
enter a number5
enter another number4.999
Close
```

QUESTION NO: 8

AIM: Write a program that asks the user to enter a word and prints out whether that word contains any vowels.

DESCRIPTION:

program that asks the user to enter a word and prints out whether that word contains any vowels.

ALGORITHM:

STEP 1: START.

STEP 2: read word.

STEP 3: vowels = "aeiouAEIOU"

STEP 4: initialise variables lag = 0

count = 0.

STEP 5: using a loop that runs on each word of the word.

STEP6: nest another loop that run on the vowels.

STEP 7: for each letter in word variable check whether the letter is found in any letter in vowels.

STEP 8: if yes, print the word contains vowels.

STEP 9: if no continue checking the next letters in the word.

STEP 10: if no letter is a vowel from word. Print word contains no vowels.

STEP 11: END.

PROGRAM:


```
word = input("ENTER A WORD :")
```

```
vowels = "aeiouAEIOU"
```

```
flag = 0
```

```
count = 0
```

```
for i in word:
```

```
    for j in vowels:
```

```
        if j==i:
```

```
            flag = 1
```

```
            count = count + 1
```

```
            break
```

```
if flag==1:
```

```
    print("The given word conatains vowels")
```

```
    break
```

else:

continue

if count==0:

print("The given word does not contain any vowels")

OUTPUT:

```
word = input("ENTER A WORD :")
vowels = "aeiouAEIOU"
flag = 0
count = 0
for i in word:
    for j in vowels:
        if j==i:
            flag = 1
            count = count +1
            break
    if flag==1:
        print("The given word conatains vowels")
        break
    else:
        continue
if count==0:
    print("The given word does not contain any vowels")
```

```
ENTER A WORD :Aarthi
The given word conatains vowels
```

QUESTION NO: 9

AIM: Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

DESCRIPTION:

Program takes two strings generates a new string which consists words from given 2 strings placed alternatively.

ALGORITHM:

STEP 1: START.

STEP 2: read string 1 & 2.

STEP 3: initialise string s3 as empty string.

STEP 4: find lengths of two strings and store in l1 and l2 using len() method.

STEP 5: if length of the strings is not equal another exit the program.

STEP6: else using a for loop and range method iterate from 0 to l1-1 .

STEP 7: for rach iteration append the two strings and store in string 3 with respective indexes.

STEP 8: print string 3.

STEP 9: END.

PROGRAM:

```
str1 = input("Input a string :")
```

```
str2 = input("Input one more string :")
```

```
str3 = ""
```

```
l1 , l2 = len(str1) , len(str2)
```

```
if l1 != l2:
```

```
    print("Operation not possible ! \n" + "Input  
strings of equal length")
```

```
else:
```

```
    for i in range(0,l1):
```

```
        str3 += (str2[i]+str1[i])
```

```
    print(str3)
```

OUTPUT:

```
str1 = input("Input a string :")
str2 = input("Input one more string :")
str3 = ""
l1 , l2 = len(str1) , len(str2)
if l1 != l2:
    print("operation not possible ! \n"+"Input strings of equal length")
else:
    for i in range(0,l1):
        str3 += (str2[i]+str1[i])
    print(str3)
```

Input a string :abcde
Input one more string :ABCDE
AaBbCcDdEe

QUESTION NO: 10

AIM:) Write a program that asks the user for a large integer and inserts commas into it

according to the standard American convention for commas in large numbers. For

instance, if the user enters 1000000, the output should be 1,000,000.

DESCRIPTION:

program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers.

ALGORITHM:

STEP 1: START.

STEP 2: Read the input from user and cast it of int type and store in a variable num.

STEP 3: using format method with a special modification { :, } to generate the desired output.

STEP 4: print the generated output.

STEP 5: END.

PROGRAM:

```
num = int(input("Input any whole number :"))
```

```
print("{:,}".format(num))
```

OUTPUT:


```
|: num = int(input("Input any whole number :"))  
print("{:,}".format(num))
```

Input any whole number :1000000
1,000,000

QUESTION NO: 11

AIM: In algebraic expressions, the symbol for multiplication is often left out, as in $3x+4y$ or

$3(x+5)$. Computers prefer those expressions to include the multiplication symbol, like

$3*x+4*y$ or $3*(x+5)$. Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate.

DESCRIPTION:

Program for adding $*$ at required places of an input given as an linear algebraic expression.

ALGORITHM:

STEP 1: START.

STEP 2: read a string of algebraic expression.

STEP 3: type caste the string into list.

STEP 4: initialise the result variable as a empty string. And a variable $l = 0$.

STEP 5: while l lies less than the length of the string.

STEP6: if $l[i] == '('$

Index = i.index(')')

STEP 7: $s2 = " ".join([i:index=1])$

Result += '*' + s2

STEP 8: $l = l + len(s2)$

STEP 9: elif:

L[i].isalpha():

Result = result + "*" + l[i]

l = i+1

STEP 10: else:

Result = result + l[i]

l = i+1

Step 11: print result.

Step 12: END.

PROGRAM:

**s = input('input an linear algebraic
expression : ')**

l = list(s)

```
result = ""
```

```
i = 0
```

```
while i < len(s):
```

```
    if l[i] == '(':
```

```
        index = l.index(')')
```

```
        s2 = "".join(l[i:index+1])
```

```
        result = result + "*" + s2
```

```
        i = i + len(s2)
```

```
    elif l[i].isalpha():
```

```
        result = result + "*" + l[i]
```

```
        i = i + 1
```

```
    else:
```

```
        result = result + l[i]
```

`i = i+1`

`print(result)`

OUTPUT:

```
s = input('input an linear algebraic expression : ')
l = list(s)
result = ""
i = 0
while i < len(s):
    if l[i] == '(':
        index = l.index(')')
        s2 = "".join(l[i:index+1])
        result = result + "*" + s2
        i = i + len(s2)
    elif l[i].isalpha():
        result = result + "*" + l[i]
        i = i + 1
    else:
        result = result + l[i]
        i = i + 1
print(result)
```

input an linear algebraic expression : 3(x+5)
3*(x+5)

QUESTION NO: 12

AIM: Write a program that generates a list of 20 random numbers between 1 and 100.

(a) Print the list.

(b) Print the average of the elements in the list.

(c) Print the largest and smallest values in the list.

(d) Print the second largest and second smallest entries in the list

(e) Print how many even numbers are in the list.

DESCRIPTION:

Python program that generates a list of 20 random numbers between 1 and 100 using random module and performs specified operations.

ALGORITHM:

STEP 1: START.

STEP 2: from random module import randint.

STEP 3: initialise a variable li as empty list.

STEP 4: for i in range of 1 to 20 run a loop.

STEP 5: append a random number in the range of 1 to 99 into the list li created.

STEP 6: print li

STEP 7: print average of elements in li using sum method and len() .

STEP 8: in order to get a list of unrepeated values from the list , use dict.fromkeys() metho with argument li and obtain a list by type casting it into list.

STEP 9: sort the new list using .sort() method.

STEP 10: access the elements by using normal and reverse indexations.

Step 11: using a loop go through each element of the list and check if no is divisible by 2 , and increment count variable previously initialised to 0 at every step.

Step 12: print(count).

Step 13: END.

PROGRAM:

```
from random import randint
```

```
li = []
```

```
for i in range(1,21):
```

```
    li.append(randint(2,99))
```

```
#a
```

```
print(li)
```

```
#b
```

```
l = len(li)
```

```
s = sum(li)
```

```
avg = s/l
```

```
print(avg)
```

```
#c
```

```
li.sort()
```

```
li1 = list(dict.fromkeys(li)) #dict.fromkeys()
```

```
return dictionary with key values
```

```
l1 = len(li1)          # since dictionaries
```

```
donot contain repeated keys we get a list of  
non repeated values by type casting the list
```

```
print("The largest number in the list is : ",li[-  
1])
```

```
print("The smallest number in the list is :  
",li[0])
```

```
#d
```

```
print("The second largest number in the list  
is : ",li[-2])
```

```
print("The second smallest number in the list  
is : ",li[1])
```

```
#e
```

```
count = 0
```

```
for i in li:
```

```
    if i%2==0:
```

```
        count = count+1
```

```
print(count)
```

OUTPUT:

```
#a  
print(li)
```

```
[57, 35, 30, 23, 24, 94, 93, 67, 64, 68, 94, 42, 27, 30, 83, 88, 31, 46, 77, 25]
```

```
: #b
l = len(li)
s = sum(li)
avg = s/l
print(avg)
```

54.9

```
: #c
li.sort()
li1 = list(dict.fromkeys(li)) #dict.fromkeys() return dictionary
l1 = len(li1) # since dictionaries donot contain repeated keys we get a list of no
print("The largest number in the list is : ",li[-1])
print("The smallest number in the list is : ",li[0])
```

The largest number in the list is : 94
The smallest number in the list is : 23

```
: #d
print("The second largest number in the list is : ",li[-2])
print("The second smallest number in the list is : ",li[1])
```

The second largest number in the list is : 94
The second smallest number in the list is : 24

```
: #e
count = 0
for i in li:
    if i%2==0:
        count = count+1
print(count)
```

10

QUESTION NO: 13

AIM: Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.

DESCRIPTION:

program that asks the user for an integer and creates a list that consists of the factors of that integer.

ALGORITHM:

STEP 1: START.

STEP 2: take input of int type.

STEP 3: create a empty list [].

STEP 4: for I in range of 1 to n+1

Check if $n \% i == 0$

If yes , append l to l.

STEP 5: if no, continue to next iteration.

STEP6: print l which contains factors of n.

STEP 7: END.

PROGRAM:

```
n = int(input("Enter a number : "))
```

```
l=[]
```

```
for i in range(1,n+1):
```

```
    if n%i==0:
```

```
        l.append(i)
```

```
print("Factors of given number are : ",l)
```

OUTPUT:

```
: n = int(input("Enter a number : "))  
l=[]  
for i in range(1,n+1):  
    if n%i==0:  
        l.append(i)  
print("Factors of given number are : ",l)
```

Enter a number : 14

Factors of given number are : [1, 2, 7, 14]

QUESTION NO: 14

AIM: Write a program that generates 100 random integers that are either 0 or 1.

Then find

the longest run of zeros, the largest number of zeros in a row. For instance, the longest

run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.

DESCRIPTION:

program that generates 100 random integers that are either 0 or 1. Then finds the longest run of zeros, the largest number of zeros in a row.

ALGORITHM:

STEP 1: START.

STEP 2: import randint from random.

STEP 3: create a empty list l.

STEP 4: for l in range of 0 to 100

In each iteration append random integer generated of range 0 and 1 .

STEP 5: initialise variables count,
max_count = 0.

STEP6: for l in L

If l==0 , then count is incremented by 1.

STEP 7: else if count is greater than
max_count , max count is updated with
count & count is made to 0.

STEP 8: after outside loop is completed
print max_count which stored the longest
run of 0's in the list generated.

STEP 9: END.

PROGRAM:

```
from random import randint
```

```
l = []
```

```
for i in range(100):
```

```
    l.append(randint(0,1)) #generating list of  
    1's and 0's randomly
```

```
count,max_count = 0,0
```

```
for i in l:
```

```
    if i==0:
```

```
        count+=1
```

```
    else:
```

```
        if count>max_count:
```

max_count = count

count = 0

else:

print(l)

print(max_count)

OUTPUT:

```
from random import randint
l = []
for i in range(100):
    l.append(randint(0,1)) #generating list of 1's and 0's randomly
count,max_count = 0,0
for i in l:
    if i==0:
        count+=1
    else:
        if count>max_count:
            max_count = count
        count = 0
else:
    print(l)
    print(max_count)
```

```
[1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0,
0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0,
0, 0, 0, 1]
```

5

QUESTION NO: 15

AIM: Write a program that removes any repeated items from a list so that each item appears

at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

DESCRIPTION:

program that removes any repeated items from a list so that each item appears at most once.

ALGORITHM:

STEP 1: START.

STEP 2: create a list that consists of random numbers.

STEP 3: type cast the list into set or using list(dict.fromkeys) method to obtain a new list that do not contain repeated elements.

STEP 5: print the new list.

STEP6: END.

PROGRAM:

```
li = [1,1,2,3,4,3,0,0]
```

```
li1 = list(dict.fromkeys(li))
```

```
print(li1)
```

OUTPUT:

```
li = [1,1,2,3,4,3,0,0]
li1 = list(dict.fromkeys(li))
print(li1)
```

```
[1, 2, 3, 4, 0]
```

QUESTION NO: 16

AIM: Write a program that asks the user to enter a length in feet. The program should then

give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.

DESCRIPTION:

program that asks the user to enter a length in feet. And shows output of in specified length conversion by choice selected by the user.

ALGORITHM:

STEP 1: START.

STEP 2: take input for length in feet.

STEP 3: based on choice of the user print the predefined calculations .

STEP 4: END.

PROGRAM:

```
feet = int(input("Enter a length in feet: "))
```



```
print("""choose 1 to convert into  
inches,\nchoose 2 to convert into  
yards,\nchoose 3 to convert into  
miles,\nchoose 4 to convert into  
millimeters,\nchoose 5 to convert into  
centimeters,\nchoose 6 to convert into  
meters,\nchoose 7 to convert into  
kilometers""")
```

```
integer = int(input())
```

```
inches = feet * 12
```

```
yards = feet * 0.33333
```

```
miles = feet * 0.000189393939
```

```
millimeters = feet * 304.8
```

```
centimeters = feet * 30.48
```

```
meters = feet * 0.3048
```

kilometers = feet * 0.0003048

convert =

[feet,inches,yards,miles,millimeters,centimeters,
meters,kilometers]

print(convert[integer])

OUTPUT:

```
: feet = int(input("Enter a length in feet: "))
print("""choose 1 to convert into inches,\nchoose 2 to convert into yards,\nchoose 3 to convert into miles,\nchoose 4 to convert into millimeters,\nchoose 5 to convert into centimeters,\nchoose 6 to convert into meters,\nchoose 7 to convert into kilometers""")
integer = int(input())

inches = feet * 12
yards = feet * 0.33333
miles = feet * 0.000189393939
millimeters = feet * 304.8
centimeters = feet * 30.48
meters = feet * 0.3048
kilometers = feet * 0.0003048

convert = [feet,inches,yards,miles,millimeters,centimeters,meters,kilometers]

print(convert[integer])
```

```
Enter a length in feet: 12
choose 1 to convert into inches,
choose 2 to convert into yards,
choose 3 to convert into miles,
choose 4 to convert into millimeters,
choose 5 to convert into centimeters,
choose 6 to convert into meters,
choose 7 to convert into kilometers
2
3.99996
```

QUESTION NO: 17

AIM: Write a function called `sum_digits` that is given an integer `num` and returns the sum of the digits of `num`.

DESCRIPTION:

Program consisting of a function that return the sum of digits of number given a number.

ALGORITHM:

STEP 1: START.

STEP 2: read `a`.

STEP 3: call function `sum` with argument as `a`

STEP 4: in `sum` function type cast and store the value passed as a list.

STEP 5: initialise sum to 0.

STEP6: for a variable l in y ,

Sum += int(i)

STEP 7: return sum.

STEP 8: print sum

STEP 9: END.

PROGRAM:

```
def sum_digits(x):
```

```
    y = list(x)
```

```
    sum = 0
```

```
    for i in y:
```

```
        sum+=int(i)
```

```
    return sum
```

```
a = input("Input any number : ")
```

```
print("The sum of the digits of the given  
number is :",sum_digits(a))
```

OUTPUT:

```
def sum_digits(x):  
    y = list(x)  
    sum = 0  
    for i in y:  
        sum+=int(i)  
    return sum  
  
a = input("Input any number : ")  
print("The sum of the digits of the given number is :",sum_digits(a))
```

```
Input any number : 143  
The sum of the digits of the given number is : 8
```

QUESTION NO: 18

AIM: Write a function called `first_diff` that is given two strings and returns the first location

in which the strings differ. If the strings are identical, it should return -1.

DESCRIPTION:

Python programme consisting of a function called `first_diff` that is given two strings and returns the first location

in which the strings differ. If the strings are identical, it should return -1.

ALGORITHM:

STEP 1: START.

STEP 2: take input for two strings a & b.

STEP 3: call the first_diff function in print statement with s1 and s2 as it's arguments.

STEP 4: for l range(0,l1)

if s1[i]==s2[i]:

continue

STEP 5: elif s1[i]<s2[i]:

return i

STEP6: else:

return i

STEP 7: after all iterations of for loop is performed. If no statement was hit.

STEP 8: return -1.

STEP 9: END.

PROGRAM:

```
def first_diff(s1,s2):  
    l1 = len(s1)  
    l2 = len(s2)  
    for i in range(0,l1):  
        if s1[i]==s2[i]:  
            continue  
        elif s1[i]<s2[i]:  
            return i  
        else:  
            return i  
    else:  
        return -1  
  
s1 = input("input a string : ")
```



```
s2 = input("input one more string : ")
```

```
print(first_diff(s1,s2))
```

OUTPUT:

```
def first_diff(s1,s2):  
    l1 = len(s1)  
    l2 = len(s2)  
  
    for i in range(0,l1):  
        if s1[i]==s2[i]:  
            continue  
        elif s1[i]<s2[i]:  
            return i  
        else:  
            return i  
    else:  
        return -1  
  
s1 = input("input a string : ")  
s2 = input("input one more string : ")  
  
print(first_diff(s1,s2))
```

```
input a string : raju  
input one more string : raju  
-1
```

QUESTION NO: 19

AIM: Write a function called `number_of_factors` that takes an integer and returns how many factors the number has.

DESCRIPTION:

Python programme consisting of a function named `number_of_factors` that takes an integer and returns how many factors the number has.

ALGORITHM:

STEP 1: START.

STEP 2: `number_of_factors` function consists a parameter `x`.

STEP 3: initialise factor to a empty list.

STEP 4: for I in range(1,x+1)

If $x \% i == 0$:

Append I to factors.

STEP 5: else:

continue

STEP6: if all the iterations are done return factors.

STEP 7: store the input as casted into integer type.

STEP 8: print the returned value after function is called.

STEP 9: END.

PROGRAM:

```
def number_of_factors(x):  
    factors = []  
    for i in range(1,x+1):  
        if x%i==0:  
            factors.append(i)  
        else:  
            continue  
    else:  
        return factors  
  
y = int(input("Please input a number:"))  
print(number_of_factors(y))
```

OUTPUT:

```
: def number_of_factors(x):  
    factors = []  
    for i in range(1,x+1):  
        if x%i==0:  
            factors.append(i)  
        else:  
            continue  
    else:  
        return factors  
  
y = int(input("Please input a number:"))  
print(number_of_factors(y))
```

Please input a number:24
[1, 2, 3, 4, 6, 8, 12, 24]

QUESTION NO: 20

AIM: Write a function called `is_sorted` that is given a list and returns `True` if the list is sorted and `False` otherwise.

DESCRIPTION:

Program consisting of a function called `is_sorted` that is given a list and returns `True` if the list is sorted and `False` otherwise.

ALGORITHM:

STEP 1: START.

STEP 2: read n.

STEP 3: for l in range of 0 to n take inputs and append them to a empty list previously declared.

STEP 4: after taking inputs pass the list to the function is_sorted(y).

STEP 5: is_sorted checks the passed list against the sorted form the same list .

STEP6: if both are identical it return true.

STEP 7: else returns false.

STEP 8: END.

PROGRAM:

```
def is_sorted(x):
```

```
    if x == sorted(x):
```

```
        return True
```

else:

return False

y = []

n=int(input("Please input the size of the list :"))

print("Please input the elements :")

for i in range(0,n):

y.append(int(input()))

else:

print(is_sorted(y))

OUTPUT:


```
def is_sorted(x):  
    if x==sorted(x):  
        return True  
    else:  
        return False  
  
y = []  
n=int(input("Please input the size of the list :"))  
print("Please input the elements :")  
for i in range(0,n):  
    y.append(int(input()))  
else:  
    print(is_sorted(y))
```

Please input the size of the list :5

Please input the elements :

1

2

3

4

5

True

QUESTION NO: 21

AIM: Write root that a function called is given a number x and an integer n and returns $x^{1/n}$.

In the function definition, set the default value of n to 2.

DESCRIPTION:

Program consisting of that a function called root is given a number x and an integer n and returns $x^{1/n}$.

In the function definition, set the default value of n to 2.

ALGORITHM:

STEP 1: START.

STEP 2: read n, x .

STEP 3: call root function inside the print statement.

STEP 4: root method has default value of x and changes if passed a value for that.

STEP 5: root of the number is calculated using $x^{1/n}$.

STEP 6: returned value is displayed to the user.

STEP 7: END.

PROGRAM:

```
def root(x,n=2):
```

```
    return x**(1/n)
```

```
x = int(input("Enter a number :"))
```

```
n = int(input("Enter the nth root to be found :"))
```

```
print(root(x,n))
```

```
print(root(x))    #if no value of n is passed  
deafult value is executed as 2
```

OUTPUT:

```
: def root(x,n=2):  
    return x**(1/n)  
  
x = int(input("Enter a number :"))  
n = int(input("Enter the nth root to be found :"))  
print(root(x,n))  
print(root(x))    #if no value of n is passed deafult value is executed as 2
```

```
Enter a number :8  
Enter the nth root to be found :3  
2.0  
2.8284271247461903
```

QUESTION NO: 22

AIM:) Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.

DESCRIPTION:

Python program consisting a function called primes that is given a number n and returns a list of the first n primes.

ALGORITHM:

STEP 1: START.

STEP 2: read n.

STEP 3: pass n to primes function.

STEP 4: primes function has a parameter x by default set to 100.

STEP 5: initialized a list li as an empty list.

STEP6: for l in range of 2 to 5000

Use another nested for loop in range of 2 to i.

STEP 7: if $i \% j == 0$:

then break the nested for.

STEP 8: else:

Append l to the list.

STEP 9: for the outer for loop, set else condition to print list generated from index 0 to x(excluded in slicing).

STEP 10: END.

PROGRAM:

```
def primes(x=100):
```

```
    li = []
```

```
    for i in range(2,5000):
```

```
        for j in range(2,i):
```

```
            if (i%j==0):
```

```
                break
```

```
        else:
```

```
            li.append(i)
```

```
    else:
```

```
        print(li[0:x])
```

```
n = int(input("Input a number : "))
```

primes(n)

primes() #by default x is set to 100

OUTPUT:

```
def primes(x=100):  
    li = []  
    for i in range(2,5000):  
        for j in range(2,i):  
            if (i%j==0):  
                break  
            else:  
                li.append(i)  
        else:  
            print(li[0:x-1])  
  
n = int(input("Input a number : "))  
primes(n)  
primes() #by default x is set to 100
```

Input a number : 50

[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, 211, 223, 227]
[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, 211, 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, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523]

QUESTION NO: 23

AIM: Write a function called merge that takes two already sorted lists of possibly different

lengths, and merges them into a single sorted list.

(a) Do this using the sort method. (b) Do this without using the sort method.

DESCRIPTION:

Python Program for a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list.

(a) Do this using the sort method. (b) Do this without using the sort method.

ALGORITHM:

STEP 1: START.

STEP 2: define function merge having two parameters l1 and l2.

STEP 3: when two lists are passed as arguments to the function , the functions appends two lists passed using + addition operation.

STEP 4: using .sort() on the appended list we can print the merged and sorted list.

STEP 5: we can also sort the merged list using merge sort function.

STEP6: print the returned list.

STEP 7: END.

PROGRAM:

#a using sort method

def merge(L1,L2): # also works by taking
inputs for the list using loops

L3 = (L1 + L2)

print("Without using sort :",merge1(L3))

L3.sort()

print("With sort :",L3)

#b without using sort method

def merge1(L): # using selection sort

for i in range(0,len(L)-1):

for j in range(i+1,len(L)):

if L[i] > L[j]:

temp = L[j]

$L[j] = L[i]$

$L[i] = \text{temp}$

else:

return L

$l1 = [1, 2, 3, 4, 5, 6, 7, 8,]$

$l2 = [18, 17, 16, 15, 14, 13, 12, 11]$

$\text{merge}(l1, l2)$

OUTPUT:

```
#a usig sort method
def merge(L1,L2): # also works by taking inputs for the List using Loops
    L3 = (L1+L2)
    print("Without using sort :",merge1(L3))
    L3.sort()
    print("With sort :",L3)
```

```
#b without using sort method
```

```
def merge1(L): # using selection sort
    for i in range(0,len(L)-1):
        for j in range(i+1,len(L)):
            if L[i]>L[j]:
                temp = L[j]
                L[j] = L[i]
                L[i] = temp
    else:
        return L
```

```
l1 = [1,2,3,4,5,6,7,8,]
l2 = [18,17,16,15,14,13,12,11]
```

```
merge(l1,l2)
```

```
Without using sort : [1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18]
With sort : [1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18]
```

QUESTION NO: 24

AIM: Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word.

DESCRIPTION:

program that asks the user for a word and finds all the smaller words that can be made from the letters of that word.

ALGORITHM:

STEP 1: START.

STEP 2: from random import *

STEP 3: read l

STEP 4: for l in range of 2 to len(l)+1

**STEP 5: nest another loop where j in
range(2,len(l)+1)**

STEP6: print sample(l,i)

STEP 7: END.

PROGRAM:

From random import *

L = input("Enter a word :")

For l in range(2,len(l)+1):

For j in range(2,len(l)+1):

Print(sample(l,i))

OUTPUT:

```
from random import *
l = input("Enter a word : ")
for i in range(2,len(l)+1):
    for j in range(2,len(l)+1):
        print(sample(l,i))
```

Enter a word : Sunny

```
['n', 'y']
['S', 'u']
['u', 'n']
['n', 'S']
['y', 'n', 'n']
['y', 'n', 'n']
['S', 'y', 'n']
['n', 'u', 'y']
['n', 'y', 'S', 'n']
['u', 'n', 'S', 'y']
['S', 'n', 'n', 'u']
['n', 'y', 'S', 'n']
['S', 'u', 'n', 'y', 'n']
['n', 'n', 'S', 'y', 'u']
['u', 'S', 'n', 'y', 'n']
['S', 'y', 'u', 'n', 'n']
```