

Week 4- Programs on Iterative constructs, Lists and Tuples

Name: NAREN CHANDRASHEKHAR

SRN: PES2UG20CS216

Section: G

Program 1	<ul style="list-style-type: none"> a) Write a program to generate fibonacci series till n terms b) Find factorial of a number c) prints all prime numbers from 2 - n
	<p>Algorithm:</p> <p>For Fibonacci Series</p> <p>Step1: Start</p> <p>Step2: Input the value of number of terms, n</p> <p>Step3: Set value of fib1 and fib2 as 0 and 1 respectively</p> <p>Step4: Set counter value as 3</p> <p>Step5: fib3 = fib1+fib2</p> <p>Step6: Print fib3</p> <p>Step7: fib1 = fib2</p> <p>Step8: fib2 = fib3</p> <p>Step9: counter=counter+1</p> <p>Step10: Repeat step5 to 9 till value of counter is <=n</p> <p>For Factorial</p> <p>Step1: Start</p> <p>Step2: Input number to find factorial, m</p> <p>Step3: Set counter, j and factorial as 1</p> <p>Step4: factorial=factorial*j</p> <p>Step5: j=j+1</p> <p>Step6: while j<=m repeat steps 4 and 5</p> <p>Step7: Print factorial</p> <p>Step8: End</p> <p>For Prime numbers between 2 and p</p> <p>Step1: Start</p> <p>Step2: Input value of p</p> <p>Step3: num = 2;</p> <p>Step4: k=2; isPrime=True</p> <p>Step5: if num%k == 0:</p>

Week 3: <title>

	<p>isPrime = False; go to step7</p> <p>Step6: k=k+1</p> <p>Step7: Repeat steps 5 and 6 till k<num</p> <p>Step8: Print the prime number, num = num + 1</p> <p>Step9: Repeat steps 4 to 8 till num < p</p> <p>Step9: End</p>
	<p>Program with comments:</p> <p>#Fibonacci Series</p> <p>n = int(input("Enter the number of terms in Fibonacci series "))</p> <p>fib1 = 0 #assigning first term in Fibonacci as 0</p> <p>fib2 = 1 #assigning first term in Fibonacci as 1</p> <p>print(fib1)</p> <p>print(fib2)</p> <p>i = 3 #Set counter as 3 as two terms are already printed</p> <p>while i<=n:</p> <p> fib3 = fib1+fib2 #assign next fibonacci term as the sum of previous two terms</p> <p> print(fib3)</p> <p> #changing values of fib1 and fib2 to the next values</p> <p> fib1 = fib2</p> <p> fib2 = fib3</p> <p> i+=1</p> <p>#Factorial of a number</p> <p>m = int(input("Enter a positive number to find its factorial "))</p> <p>j = 1 #counter</p> <p>factorial = 1 #assigning factorial as 1, as 0 and 1 factorial is 1</p> <p>while j<=m:</p> <p> factorial = factorial*j #calculating factorial</p> <p> j+=1</p> <p>print("The factorial of the number is",factorial)</p> <p>#Prime numbers</p> <p>p = int(input("Enter a positive number to print all prime number from 2 "))</p> <p>print("Prime numbers between 2 to",p,"are")</p>

Week 3: <title>

	<pre> for num in range(2,p): k = 2 isPrime = True #assuming all numbers to be prime while k<num: if num%k == 0: isPrime = False #segregating prime numbers from non prime numbers break k+=1 if(isPrime): print(num) </pre>
	<p>Output:</p> <pre> D:\PES\Semester 1\Computer Science- Python Programming\PythonLab\week4>program1.py Enter the number of terms in Fibonacci series 8 0 1 1 2 3 5 8 13 Enter a positive number to find its factorial 7 The factorial of the number is 5040 Enter a positive number to print all prime number 15 Prime numbers between 2 to 15 are 2 3 5 7 11 13 </pre>
Program 2	<p>Write a python program to perform the following operations using given list as input:</p> <ol style="list-style-type: none"> Given a heterogenous list, create separate lists for different types of data. Write a python program to achieve the same. Sort in ascending and descending order <ol style="list-style-type: none"> list of strings list of number
	<p>Algorithm:</p> <p>Step1: Start</p> <p>Step2: Set, l1=[1,2,"hello",(5,8,9),10.5,[6,7],3.5,(1,2,3),[11,12],{23,34},"aaa"]</p> <p>Step3: Create empty lists l_int, l_float, l_str, l_tuple, l_list, l_set</p> <p>Step4: set i = 0</p>

Week 3: <title>

	<p> Step5: if type of l1[i] = int, add l1[i] to l_int[] Step6: elif type of l1[i] = float, add l1[i] to l_float[] Step7: elif type of l1[i] = str, add l1[i] to l_str[] Step8: elif type of l1[i] = tuple, add l1[i] to l_tuple[] Step9: elif type of l1[i] = list, add l1[i] to l_list[] Step10: elif type of l1[i] = set, add l1[i] to l_set[] Step11: i=i+1 Step12: Repeat steps 5 to 11 till end of the list Step13: End </p> <p> Sort in ascending and descending order Step1: Start Step2: Initialise strlist=['aaa','xxx','bbb','abs','xyz','bcd'] numlist=[23,12,34,56,26,33,22] Step3: strlist.sort() Step4: Print strlist Step5: strlist.sort(reverse=True) Step6: Print strlist Step7: numlist.sort() Step8: Print numlist Step9: numlist.sort(reverse=True) Step10: print numlist Step11: End </p>
	<p> Program with Comments: #A l1=[1,2,"hello",(5,8,9),10.5,[6,7],3.5,(1,2,3),[11,12],{23,34},"aaa"] </p> <p> #creating empty lists for every data type l_int = [] l_float = [] l_str = [] l_tuple = [] l_list = [] l_set = [] </p> <p> #checking data type of elements for i in l1: c = type(i) if(c == int): l_int.append(i) elif(c == float): l_float.append(i) </p>

Week 3: <title>

	<pre> elif(c == str): l_str.append(i) elif(c == tuple): l_tuple.append(i) elif(c == list): l_list.append(i) elif(c == set): l_set.append(i) print("List of integers",l_int) print("List of floating point numbers",l_float) print("List of strings",l_str) print("List of tuples",l_tuple) print("List of lists",l_list) print("List of sets",l_set) #B strlist=['aaa','xxx','bbb','abs','xyz','bcd'] numlist=[23,12,34,56,26,33,22] #organising list in ascending and descending order respectively strlist.sort() print("The string in ascending order is",strlist) strlist.sort(reverse=True) print("The string in descending order is",strlist) numlist.sort() print("The numbers in ascending order is",numlist) numlist.sort(reverse=True) print("The numbers in descending order is",numlist) </pre>
	<p>Output:</p>

Week 3: <title>

	<pre> D:\PES\Semester 1\Computer Science- Python Programming\PythonLab\Week4>program2_lists.py List of integers [1, 2] List of floating point numbers [10.5, 3.5] List of strings ['hello', 'aaa'] List of tuples [(5, 8, 9), (1, 2, 3)] List of lists [[6, 7], [11, 12]] List of sets [{34, 23}] The string in ascending order is ['aaa', 'abs', 'bbb', 'bcd', 'xxx', 'xyz'] The string in descending order is ['xyz', 'xxx', 'bcd', 'bbb', 'abs', 'aaa'] The numbers in ascending order is [12, 22, 23, 26, 33, 34, 56] The numbers in descending order is [56, 34, 33, 26, 23, 22, 12] </pre>
Program 3	<p>Generate heart rate randomly between 50 to 120 at time interval of 3 hours for 24 hours.</p> <ol style="list-style-type: none"> If heart rate is between 50-65 print as bradycardia(lower heart rate) if greater than 100 print as tachycardia(higher heart rate). Else print as normal. Count number of Bradycardia and tachycardia if any of this is greater than 3 display as risk. Print the maximum heart rate and minimum heart rate
	<p>Algorithm:</p> <p>Step1: Start</p> <p>Step2: Set brady_cardia = 0; tachy_cardia = 0; c = 0; heart_rate=[]</p> <p>Step3: Print 8 random values between 50 and 120</p> <p>Step4: if i>50 and i<65:</p> <p style="padding-left: 40px;">brady_cardia= brady_cardia+1</p> <p style="padding-left: 40px;">print Bradycardia</p> <p style="padding-left: 40px;">elif i>100:</p> <p style="padding-left: 40px;">tachy_cardia=tachy_cardia+1</p> <p style="padding-left: 40px;">print Tachycardia</p> <p style="padding-left: 40px;">else:</p> <p style="padding-left: 40px;">print Normal</p> <p>Step5: if (brady_cardia or tachy_cardia)>3:</p> <p style="padding-left: 40px;">Print Risk</p> <p>Step6: Print maximum and minimum heart rate</p> <p>Step7: End</p>
	<p>Program with comments:</p> <pre> import random #setting limit of Bradycardia, Trachycardia and counter as 0 brady_cardia = 0 </pre>

Week 3: <title>

	<pre> tachy_cardia = 0 c = 0 heart_rate=[] #creating empty list #getting 8 random values of heart rate between 50 and 120 while(c<8): i = random.randint(50,120) heart_rate.append(i) c=c+1 print("The heart rate at intervals of 3 hours for 24 hours is",heart_rate) for i in heart_rate: if i>50 and i<65: brady_cardia+=1 #incrementing count of Bradycardia print("Heart rate",i,"is Bradycardia") elif i>100: tachy_cardia+=1 #incrementing count of Tachycardia print("Heart rate",i,"is Tachycardia") else: print("Heart rate",i,"is Normal") if (brady_cardia or tachy_cardia)>3: print("Risk because count of Bradycardia or Tachycardia is greater than 3") #printing maximum and minimum heart rate print("The maximum heart rate is",max(heart_rate)) print("The minimum heart rate is",min(heart_rate)) </pre>
	Output:

Week 3: <title>

	<pre> D:\PES\Semester 1\Computer Science- Python Programming\PythonLab\Week4>program3_hearttrate.py The heart rate at intervals of 3 hours for 24 hours is [116, 110, 117, 59, 56, 67, 60, 60] Heart rate 116 is Tachycardia Heart rate 110 is Tachycardia Heart rate 117 is Tachycardia Heart rate 59 is Bradycardia Heart rate 56 is Bradycardia Heart rate 67 is Normal Heart rate 60 is Bradycardia Heart rate 60 is Bradycardia Risk because count of Bradycardia or Tachycardia is greater than 3 The maximum heart rate is 117 The minimum heart rate is 56 </pre>
Program 4	<p>Enter marks of students till you need to stop.</p> <ol style="list-style-type: none"> Find maximum marks Find number of students who have scored highest Find second highest marks Enter fail marks and remove if fail marks present in list
	<p>Algorithm:</p> <p>Step1: Start</p> <p>Step2: Initialize student_marks = []; n = 0; count = 0</p> <p>Step 3: Read the marks in student_marks[] until the input is negative value</p> <p>Step 4: Print max(student_marks)</p> <p>Step 5: max_marks = max(student_marks)</p> <p>Step 6: if student_marks[n] == max_marks: count+=1</p> <p>Step 7: Repeat step 6 for each element in student_marks</p> <p>Step 8: print - "The number of students who have scored highest marks",count</p> <p>Step 9: new_list =[]</p> <p>Step 10: if student_marks[n] != max_marks:</p> <p style="padding-left: 40px;">Add marks to new_list</p> <p>Step 11: Repeat step 10 for all elements in student_marks</p> <p>Step 12: max_marks2 = max(new_list)</p> <p>Step 13: print("The second highest marks is",max_marks2)</p> <p>Step 14: Read the value of fail_marks</p> <p>Step 15: pass_list = [] #creating new list for pass students</p> <p>Step 16: for each element n in student_marks:</p> <p style="padding-left: 40px;">if n>fail_marks:</p> <p style="padding-left: 80px;">pass_list.append(n)</p> <p>Step 17: print("The marks of students who have passed are",pass_list)</p> <p>Step 18: End</p>
	Program with comment:

Week 3: <title>

	<pre> student_marks = [] n = 0 count = 0 print("Enter the marks of each student for a given subject(to stop enter a negative number) ") #input value of marks while (n>=0): n = int(input()) if n>=0: student_marks.append(n) print(student_marks) print("The highest marks scored in the subject is",max(student_marks)) #counting number of maximum marks by comparing and incrementing value of count max_marks = max(student_marks) for n in student_marks: if n == max_marks: count+=1 print("The number of students who have scored highest marks",count) #counting second highest marks new_list =[] #creating a new list for n in student_marks: if n != max_marks: new_list.append(n) #new list without highest marks max_marks2 = max(new_list) print("The second highest marks is",max_marks2) fail_marks = int(input("The fail marks is ")) #input failed marks pass_list = [] #creating new list for pass students for n in student_marks: if n>fail_marks: pass_list.append(n) print("The marks of students who have passed are",pass_list) </pre>
	Output:

Week 3: <title>

	<pre> D:\PES\Semester 1\Computer Science- Python Programming\PythonLab\Week4>program4_marks.py Enter the marks of each student for a given subject(to stop enter a negative number) 30 70 98 99 90 23 31 -6 [30, 70, 98, 99, 90, 23, 31] The highest marks scored in the subject is 99 The number of students who have scored highest marks 1 The second highest marks is 98 The fail marks is 35 The marks of students who have passed are [70, 98, 99, 90] </pre>
Program 5	<p>Write a python program which accepts a sequence of comma-separated values from console and generate as a list and as a tuple.</p>
	<p>Algorithm: Step1: Start Step2: Input comma separated values in x Step3: y=list(x.split(',')) Step4: Print y Step5: z=tuple(x.split(',')) Step6: Print z Step7: End</p>
	<p>Program with comment x=input("Enter the comma separated values ") print(x)</p> <p>#using split function to separate the input values y=list(x.split(',')) print("The comma separated list is",y)</p> <p>z=tuple(x.split(',')) print("The comma separated tuple is",z)</p>
	<p>Output:</p> <pre> D:\PES\Semester 1\Computer Science- Python Programming\PythonLab\Week4>program5.py Enter the comma separated values h,g,f,s,w,k,j,b,c,a,q h,g,f,s,w,k,j,b,c,a,q The comma separated list is ['h', 'g', 'f', 's', 'w', 'k', 'j', 'b', 'c', 'a', 'q'] The comma separated tuple is ('h', 'g', 'f', 's', 'w', 'k', 'j', 'b', 'c', 'a', 'q') </pre>

