**P E S INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institute under VTU, Belgaum)**

**Department of Information Science and Engineering**

**12CS308: Python Application Programming Laboratory**

**LAB ASSIGNMENT LIST**

|  |  |
| --- | --- |
| 1 a | Generate the following pattern for a given value of n. (say n = 4)  1 0 0 0  0 1 0 0  0 0 1 0  0 0 0 1 |
|  |  |
| 1 b | Create a directory PAP with 10 files. Each file should contain different info. Write a program to find biggest file in the directory PAP. |
|  |  |
| 1 c | Define a procedure, greatest, that takes as input a list of positive numbers, and returns the greatest number in that list. If the input list is empty, the output should be 0. |
|  | def greatest(p):  if(p==[]):  return 0  else:  i=0  greatest = p[i]  for a in p:  if(a>greatest):  greatest = a  return greatest  def main():  list = []  while True:  n = int(input('Enter either 0 to stop input process or a positive number to add to list: '))  if n == 0:  break  list.append(n)  print('The greatest number in the list', list, 'is', greatest(list))  main() |
|  |  |
| 2 a | Create a directory PAP with 10 files. Remove all empty files in this directory. |
|  |  |
| 2 b | Write a program to read the name and date of birth from the keyboard and insert into the table. Display the contents and verify the same at the backend also. |
|  |  |
| 2 C | Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i\*j.  Note: i=0,1.., X-1; j=0,1,¡­Y-1.  Example  Suppose the following inputs are given to the program:  3,5  Then, the output of the program should be:  [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]  Hints:  Note: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form. |
|  | input\_str = input()  dimensions=[int(x) for x in input\_str.split(',')]  rowNum=dimensions[0]  colNum=dimensions[1]  multilist = [[0 for col in range(colNum)] for row in range(rowNum)]  for row in range(rowNum):  for col in range(colNum):  multilist[row][col]= row\*col  print (multilist) |
| 3 a | Walk through this given string and find all tags.  <test>  <sub>PAP</sub>  <date>  <dd>30</dd>  <mm>9</mm>  <yy>2013</yy>  </date>  </test> |
| 3 b | Create a list of elements. Use a dict to find the frequency of each element in the list. |
| 3 c | Design a code snippet in program simply takes a list of files on the command line, and lists the contents of each, with lines numbered. |
|  | #!/usr/bin/python  # Print the contents of the files listed on the command line.  import sys  for fn in sys.argv[1:]:  try:  fin = open(fn, 'r')  except:  (type, detail) = sys.exc\_info()[:2]  print "\n\*\*\* %s: %s: %s \*\*\*" % (fn, type, detail)  continue  print "\n\*\*\* Contents of", fn, "\*\*\*"    # Print the file, with line numbers.  lno = 1  while 1:  line = fin.readline()  if not line: break;  print '%3d: %-s' % (lno, line[:-1])  lno = lno + 1  fin.close()  print |
| 4 a | Validate a given date. Check also for leap year. |
| 4 b | Write a program to find all numbers which are odd and which are palindromes between a pair of numbers. Use list comprehension. |
| 4 c | Write a program that accepts a sentence and calculate the number of letters and digits. |
|  | s = input()  d={"DIGITS":0, "LETTERS":0}  for c in s:  if c.isdigit():  d["DIGITS"]+=1  elif c.isalpha():  d["LETTERS"]+=1  else:  pass  print ("LETTERS", d["LETTERS"])  print ("DIGITS", d["DIGITS"]) |
| 5 a | Write a function to check whether a given number is prime or not. Use this function to find all prime numbers in a given range |
| 5 b | create a module MyMath.py which contains functions to compute  i) area of a sphere given the radius  ii) volume of a sphere given the radius  iii) radius of a sphere given the volume  Test these functions in the module. Write a client program to use these functions by importing. |
| 5 C | Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters. |
|  | s = input()  d={"UPPER CASE":0, "LOWER CASE":0}  for c in s:  if c.isupper():  d["UPPER CASE"]+=1  elif c.islower():  d["LOWER CASE"]+=1  else:  pass  print ("UPPER CASE", d["UPPER CASE"])  print ("LOWER CASE", d["LOWER CASE"]) |
| 6 a | Write a function which takes two arguments say a and b. Decorate to check whether a is less than b. |
| 6 b | Write a program whether a given character occurs again in the same string anywhere in the string |
| 6 C | Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:  D 100  W 200  ….  D means deposit while W means withdrawal. |
|  | import sys  netAmount = 0  while True:  s = input()  if not s:  break  values = s.split(" ")  operation = values[0]  amount = int(values[1])  if operation=="D":  netAmount+=amount  elif operation=="W":  netAmount-=amount  else:  pass  print (netAmount) |
| 7 a | Write a program to find all persons whose birth day falls in a given month.  hint: use month(dob) to get the month |
| 7 b | Find the 5 biggest files in the directory /bin. Store the result in a list. Sort the list |
| 7 C | A website requires the users to input username and password to register. Write a program to check the validity of password input by users.  Following are the criteria for checking the password:  1. At least 1 letter between [a-z]  2. At least 1 number between [0-9]  3. At least 1 letter between [A-Z]  4. At least 1 character from [$#@]  5. Minimum length of transaction password: 6  6. Maximum length of transaction password: 12  Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. |
|  | import re  value = []  items=[x for x in input().split(',')]  for p in items:  if len(p)<6 or len(p)>12:  continue  else:  pass  if not re.search("[a-z]",p):  continue  elif not re.search("[0-9]",p):  continue  elif not re.search("[A-Z]",p):  continue  elif not re.search("[$#@]",p):  continue  elif re.search("\s",p):  continue  else:  pass  value.append(p)  print (",".join(value)) |
| 8 a | Write a program which generates an HTML form which reads two numbers and calls another program (as part of action) which displays the numbers and their sum |
| 8 b | Write a program to find the # of occurrences of keywords IF DO THEN ENDDO ENDIF in a given sentence / program. |
| 8 C | Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. |
|  | freq = {} # frequency of words in text  line = raw\_input()  for word in line.split():  freq[word] = freq.get(word,0)+1  words = freq.keys()  words.sort()  for w in words:  print ("%s:%d" % (w,freq[w])) |
| 9 a | Create a class called MyStack which supports push and pop operations. |
| 9 b | Generate all perfect squares which are palindromes and so are their square roots less than or equal to given number n. |
| 9 c | 1. Write a program which uses map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]. 2. Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included). |
|  | (i) li = [1,2,3,4,5,6,7,8,9,10]  evenNumbers = map(lambda x: x\*\*2, filter(lambda x: x%2==0, li))  print (evenNumbers)  (ii) evenNumbers = filter(lambda x: x%2==0, range(1,21))  print evenNumbers |
| 10 a | Write a program to find union and intersection of given lists L1 and L2. |
| 10 b | Write a program to sum the elements of an array (list) using 4 threads. Let each thread add quarter of the array. Assume that the size of the array is a multiple of 4. |
|  |  |
| 11 a | Write a graphic based program to add two numbers and display the result. |
| 11 b | Generate the following pattern for a given value of n. (say n = 4)  1 = 1  1 + 2 = 3  1 + 2 + 3 = 6  1 + 2 + 3 + 4 = 10 |
|  |  |
| 12 a | Generate Pascal triangle. if n = 4, output should be  1  1 1  1 2 1  1 3 3 1  1 4 6 4 1 |
| 12 b | write a function which gets # of strings using variable # of arguments and returns a list of strings which are palindromes |
| 12 c | The Fibonacci Sequence is computed based on the following formula:  f(n)=0 if n=0  f(n)=1 if n=1  f(n)=f(n-1)+f(n-2) if n>1  Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console. |
|  | def f(n):  if n == 0: return 0  elif n == 1: return 1  else: return f(n-1)+f(n-2)  n=int(raw\_input())  values = [str(f(x)) for x in range(0, n+1)]  print (",".join(values)) |
|  |  |
| 13 a | Write a program to sum the elements of an array (list) using 4 processes. Let each thread add quarter of the array. Assume that the size of the array is a multiple of 4. |
| 13 b | Write a program with appropriate rules to tokenize the following as follows:  a --- b => (a, ID) (--, OP) (-, OP) (b, ID)  a << b => (a, ID) (<<, OP) (b, ID)  a < b - c => (a, ID) (< , OP) (b, ID) (-, op) (c, ID) |
| 13 c | Write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console. |
|  | def EvenGenerator(n):  i=0  while i<=n:  if i%2==0:  yield i  i+=1  n=int(raw\_input())  values = []  for i in EvenGenerator(n):  values.append(str(i))  print (",".join(values)) |
| 14 a | Given a file containing names of the state, create a file for each state. A sample file is content is given below :  karnataka:bangalore  kerala:trivandrum  kerala:munnar  tamilnad:chennai  karnataka:mysore  kerala:tekkadi  tamilnad:madurai  karnataka:hubli  kerala:kochi  tamilnad:salem  karnataka:mangalore  Hint: walk thro the file. create a dict whose key is the state name and value is the file handle. Do close all the files at the end of processing. |
| 14 b | Write a program to display the name and the age of the person. The input to the program must be date of birth and name only.  hint: check datadiff and curdate functions of mysql |
|  |  |