|  |
| --- |
| Question 1: |
|  |

|  |
| --- |
| Write a program that calculates and prints the value according to the given formula: |
|  |

|  |
| --- |
| Q = Square root of [(2 \* C \* D)/H] |
|  |

|  |
| --- |
| Following are the fixed values of C and H: |
|  |

|  |
| --- |
| C is 50. H is 30. |
|  |

|  |
| --- |
| D is the variable whose values should be input to your program in a comma-separated sequence. |
|  |

|  |
| --- |
| Example |
|  |

|  |
| --- |
| Let us assume the following comma separated input sequence is given to the program: |
|  |

|  |
| --- |
| 100,150,180 |
|  |

|  |
| --- |
| The output of the program should be: |
|  |

18,22,24

Ans

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def calculate\_formula():

C = 50

H = 30

es = []

D = [int(x) for x in input("Enter multiple values\n").split(',')]

for i in D:

Q = ((2\*C\*i/H))\*\*(0.5)

es.append(round(Q))

print(\*es, sep=",")

try:

calculate\_formula()

lg.info("Function calculate\_formula() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

|  |
| --- |
| Question 2: |
| 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]] |
|  |

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def dimension\_2d\_array():

X = int(input("Enter the value of X: "))

Y = int(input("Enter the value of Y: "))

empty\_array = []

for i in range(X):

empty\_array.append([])

for i in range(X):

for j in range(Y):

empty\_array[i].append(j\*i)

print(empty\_array)

try:

dimension\_2d\_array()

lg.info("Function dimension\_2d\_array() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

Question 3:

|  |
| --- |
| Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| without,hello,bag,world |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

bag,hello,without,world

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def words\_sort():

user\_input = [x for x in input("Enter multiple values\n").split(',')]

user\_input.sort()

print(\*user\_input, sep=",")

try:

words\_sort()

lg.info("Function words\_sort() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

Question 4:

|  |
| --- |
| Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| hello world and practice makes perfect and hello world again |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

again and hello makes perfect practice world

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def words\_sort2():

user\_input = [x for x in input("Enter multiple values\n").split(' ')]

sorted\_list = list(set(user\_input))

sorted\_list.sort()

print(\*sorted\_list, sep=" ")

try:

words\_sort2()

lg.info("Function words\_sort2() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

Question 5:

|  |
| --- |
| Write a program that accepts a sentence and calculate the number of letters and digits. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| hello world! 123 |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

|  |
| --- |
| LETTERS 10 |
|  |

DIGITS 3

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def string\_count():

check\_string = input("Enter your string here :")

strings = 0

digits = 0

spaces = 0

for i in check\_string:

if i.isdigit():

digits +=1

if i.isalpha():

strings += 1

if i == " ":

spaces += 1

print("Letters: ", strings)

print("Digits: ", digits)

try:

string\_count()

lg.info("Function string\_count() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

Question 6:

|  |
| --- |
| 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] |
|  |

|  |
| --- |
| 1. At least 1 letter between [A-Z] |
|  |

|  |
| --- |
| 3. At least 1 character from [$#@] |
|  |

|  |
| --- |
| 4. Minimum length of transaction password: 6 |
|  |

|  |
| --- |
| 5. 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. |
|  |

|  |
| --- |
| Example |
|  |

|  |
| --- |
| If the following passwords are given as input to the program: |
|  |

|  |
| --- |
| ABd1234@1,a F1#,2w3E\*,2We3345 |
|  |

|  |
| --- |
| Then, the output of the program should be: |
|  |

ABd1234@1

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def check\_password():

special\_char = "$#@"

valid\_passwords = []

declined\_passwords = []

validate = True

print("please enter list of passwords separated with a comma")

user\_input = [x for x in input("Enter multiple values\n").split(',')]

for i in user\_input:

if len(i) < 6:

validate = False

declined\_passwords.append(i)

continue

if len(i) > 12:

validate = False

declined\_passwords.append(i)

continue

if not any(j.isupper() for j in i):

validate = False

declined\_passwords.append(i)

continue

if not any(j.islower() for j in i):

validate = False

declined\_passwords.append(i)

continue

if not any(j.isdigit() for j in i):

validate = False

declined\_passwords.append(i)

continue

else:

valid\_passwords.append(i)

print("Valid passwords are :", valid\_passwords)

print("Declined passwords are :",declined\_passwords)

try:

check\_password()

lg.info("Function check\_password() has been called")

except Exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass