**DELHI PUBLIC SCHOOL BANGALORE NORTH**

**2024-2025**

**Logo

Description automatically generated**

**COMPUTER SCIENCE**

**(083)**

**PRACTICAL RECORD FILE**

**NAME:** Sriharsha

**CLASS: XII SEC** C

**Logo

Description automatically generated**

**DELHI PUBLIC SCHOOL BANGALORE NORTH**

**CERTIFICATE**

This is to certify that Sriharsha of class **XII** bearing Board Examination Roll No.: \_**\_\_\_\_\_\_\_\_\_\_\_\_** has successfully completed the **Practical Record File** during the academic year **2024-2025** in partial fulfillment of SSCE Practical Examination for the subject **COMPUTER SCIENCE (083)** conducted by **CBSE**

Date: 3/24/2024

Internal Examiner Principal Signature

External Examiner

**Computational Thinking and Programming – II**

**(PYTHON)**

**FUNCTIONS**

|  |  |
| --- | --- |
|  | Write a program to have following functions:  a) A function that takes a number as a parameter and calculates the cube for it. The function does not return a value. If there is no value passed to the function in the function call statement as an argument, the function should calculate and print the cube of 2  b) A function that takes two strings as arguments and return True if both the strings are equal otherwise returns False |
|  | Write a function that has two numeric parameters and generates a random number between those two numbers. Using this function, the main program should be able to print three numbers randomly. |
|  | Write a function that receives two string arguments and checks whether they are the same length strings (return True in this case otherwise False). Write Function Call statement and display “Same Length” or “Not Same Length” based on the result received from the function. |
|  | Write a function that takes a number n and then returns a number that has maximum face value in unit’s place. Example:   if numbers passed are 491 and 278, then the function will return 278. |
|  | Define a function named LINEARSEARCH (lst,k) that accepts a list lst and key to be searched k as parameters, Implements linear search to find all occurrences of k in lst and returns another list containing all the indexes/positions where k is found in lst. The function should return None if k is not found in lst. Write an appropriate function call statement and print the result accordingly. |
|  | Define a function that accepts a list of numbers as a parameter, find and print the sum and the average of all elements. Write an appropriate function call statement and print the result accordingly. |
|  | Define a function that takes two lists as parameters and create a third list that contains the elements from both the lists and sort it in ascending order. Write an appropriate function call statement and print the result accordingly. |
|  | Write a function part\_reverse(<list>,start, end) to reverse elements in a part of the list passed as a parameter to the function. The parameters start and end are the starting and ending index of the part of the list which must be reversed.  Assume that start < end, start>=0 and end<len(list)  Sample Input Data of List   my\_list=[1,2,3,4,5,6,7,8,9,10]  Function Call = part\_reverse(my\_list,3,6)  Output is  my\_list=[1,2,3,7,6,5,4,8,9,10] |
|  | Define a function named Unit\_Seven(\*n) that takes variable length parameters. The function should find and return a list of all values of input parameters that are ending with the digit seven. Write appropriate top level statements to call the function and to display the list obtained as result  Sample Function call:  L=Unit\_Seven(127,200,400,207,17,-127)  After the execution  L will have [127,207,17,-127] |
|  | Write a user defined function to check the validity of a password string passed as a parameter.  The function returns True if all the Validation Rules given below are satisfied otherwise it returns False.  Validation Rules:  Given password must have   * At least 1 letter between [a-z] and 1 letter between [A-Z]. * At least 1 number between [0-9]. * At least 1 character from [$#@]. * Minimum length of 5 characters and Maximum length of 15 characters.   Write appropriate top level statements to accept a password from user, check and print whether the password is Valid or Invalid by using the value returned by the above user defined function. |

**Question 1:**

Write a program to have following functions:

1. A function that takes a number as a parameter and calculates the cube for it. The function does not return a value. If there is no value passed to the function in the function call statement as an argument, the function should calculate and print the cube of 2
2. A function that takes two strings as arguments and return True if both the strings are equal otherwise returns False

Source Code:

#func to print cube of number

def cube(num):

    print(“The cube of ”, ”num” , num\*\*3)

#func to check if strings are equal or not

def check\_strings(str1, str2):

    if str1 == str2:

        return True

    else:

        return False

cube(3)

str1 = str(input(“Enter a string”))

str2 = str(input(“Enter a string:”))

print(check\_strings(str1, str2)

Output:

Enter a number:6

The cube of 6 is 216

The cube of 2 is 8

Enter a string:Play

Enter a string:clay

Given Strings are not equal

**Question 2:**

Write a function that has two numeric parameters and generates a random number between those two numbers. Using this function, the main program should be able to print three numbers randomly.

Source Code:

#generate random number between input range

import random

def random\_number(a,b):

    return random.randint(a,b)

a = int(input("Enter the lower limit: "))

b = int(input("Enter the upper limit: "))

print(random\_number(a,b))

Output:

Enter the lower limit: 12

Enter the upper limit: 45

32

**Question 3:**

Write a function that receives two string arguments and checks whether they are the same length strings (return True in this case otherwise False). Write Function Call statement and display “Same Length” or “Not Same Length” based on the result received from the function.

Source Code:

#check if same length

def check\_length(a,b):

    return len(str(a)) == len(str(b))

a = int(input("Enter the first string"))

b = int(input("Enter the second string"))

result = check\_length(a,b)

if result:

    print("The strings are of the same length")

else:

    print("The strings are not of the same length")

Output:

Enter the first string: hello

Enter the second string: borth

The strings are of the same length

**Question 4:**

Write a function that takes a number n and then returns a number that has maximum face value in unit’s place. Example:   if numbers passed are 491 and 278, then the function will return 278.

Source Code:

#return max face value in units place

def max\_face\_value(num1, num2):

    return num1 if num1%10 > num2%10 else num2

result = max\_face\_value(123,456)

print(result)

Output:

456

**Question 5:**

Define a function named LINEARSEARCH (lst,k) that accepts a list lst and key to be searched k as parameters, Implements linear search to find all occurrences of k in lst and returns another list containing all the indexes/positions where k is found in lst. The function should return None if k is not found in lst. Write an appropriate function call statement and print the result accordingly.

Source Code:

def LINEARSEARCH(lst, k):

    return [i for i, x in enumerate(lst) if x == k] or None

# Function call statement

my\_list = [3, 7, 2, 8, 2, 5, 2]

key = 2

result = LINEARSEARCH(my\_list, key)

# Print the result

print(f"The key {key} is found at indexes: {result}") if result else print(f"The key {key} is not found in the list.")

Output**:**

The key 2 is found at indexes: [2, 4, 6]

**Question 6:**

Define a function that accepts a list of numbers as a parameter, find and print the sum and the average of all elements. Write an appropriate function call statement and print the result accordingly.

Source Code:

def calculate\_sum\_and\_average(numbers):

    return sum(numbers), sum(numbers) / len(numbers)

# Function call statement

my\_numbers = [1, 2, 3, 4, 5]

sum\_result, average\_result = calculate\_sum\_and\_average(my\_numbers)

# Print the result

print(f"Sum of the numbers: {sum\_result}")

print(f"Average of the numbers: {average\_result}")

Output:

Sum of the numbers: 15

Average of the numbers: 3.0

**Question 7:**

Define a function that takes two lists as parameters and create a third list that contains the elements from both the lists and sort it in ascending order. Write an appropriate function call statement and print the result accordingly.

Source Code:

def merge\_and\_sort\_lists(list1, list2):

    return sorted(list1 + list2)

# Function call statement

list1 = [3, 1, 5]

list2 = [2, 4, 6]

result = merge\_and\_sort\_lists(list1, list2)

# Print the result

print("Merged and sorted list:", result)

Output:

Merged and sorted list: [1, 2, 3, 4, 5, 6]

**Question 8:**

Write a function part\_reverse(<list>,start, end) to reverse elements in a part of the list passed as a parameter to the function. The parameters start and end are the starting and ending index of the part of the list which must be reversed.

Assume that start < end, start>=0 and end<len(list)

Sample Input Data of List

 my\_list=[1,2,3,4,5,6,7,8,9,10]

Function Call = part\_reverse(my\_list,3,6)

Output is

my\_list=[1,2,3,7,6,5,4,8,9,10]

Source Code:

def part\_reverse(lst, start, end):

    lst[start:end+1] = lst[start:end+1][::-1]

# Sample Input Data

my\_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Function Call

part\_reverse(my\_list, 3, 6)

# Output

print(my\_list)

Output:

[1, 2, 3, 7, 6, 5, 4, 8, 9, 10]

**Question 9:**

Define a function named Unit\_Seven(\*n) that takes variable length parameters. The function should find and return a list of all values of input parameters that are ending with the digit seven. Write appropriate top level statements to call the function and to display the list obtained as result

Sample Function call:

L=Unit\_Seven(127,200,400,207,17,-127)

After the execution  L will have [127,207,17,-127]

Source Code**:**

def Unit\_Seven(\*n):

    return [x for x in n if str(x)[-1] == '7']

# Function call

L = Unit\_Seven(127, 200, 400, 207, 17, -127)

# Display the result

print(L)

Output:

[127, 207, 17, -127]

**Question 10:**

Write a user defined function to check the validity of a password string passed as a parameter.  The function returns True if all the Validation Rules given below are satisfied otherwise it returns False.

Validation Rules:

Given password must have

* At least 1 letter between [a-z] and 1 letter between [A-Z].
* At least 1 number between [0-9].
* At least 1 character from [$#@].
* Minimum length of 5 characters and Maximum length of 15 characters.

Write appropriate top level statements to accept a password from user, check and print whether the password is Valid or Invalid by using the value returned by the above user defined function.

Source Code:

def check\_password\_validity(password):

    return all([

        any(char.islower() for char in password),

        any(char.isupper() for char in password),

        any(char.isdigit() for char in password),

        any(char in '$#@' for char in password),

        5 <= len(password) <= 15

    ])

# Accept password from the user

user\_password = input("Enter your password: ")

# Check validity and print the result

if check\_password\_validity(user\_password):

    print("Password is Valid")

else:

    print("Password is Invalid")

Output:

Enter your password: Hiym@1357

Password is Valid