**Python – Practical File**

**(Subject: Computer Science)**

1. **Write a program that takes a string with multiple words and then capitalizes the first letter of each word and forms a new string out of it.**

**def capit(string\_in):**

**print("Input: " + string\_in)**

**list\_string = string\_in.split(" ")**

**final\_string = ""**

**new\_list = []**

**for i in list\_string:**

**j = i.capitalize() + " "**

**new\_list.append(j)**

**for i in new\_list:**

**final\_string = final\_string + i**

**return final\_string**

**print(capit("driver program to capitalize each word"))**

**Output:**

**Driver Program To Capitalize Each Word**

1. Write a python function to check whether the number is prime or not.

def prime\_or\_not(num):

if num < 2:

return "Please enter a positive integer greater than 1."

else:

for i in range(2, num):

if num % i == 0:

return (str(num) + " is not a prime number.")

break

else:

return (str(num) + " is a prime number.")

print(prime\_or\_not(9))

print(prime\_or\_not(13))

Output:

9 is not a prime number.

13 is a prime number.

1. Write a python function that receives two numbers and generates a random number from that range. Using this function, the main program should print three numbers randomly.

import random

def radom\_generation(low, high):

return random.randint(low, high)

for i in range(0, 3):

print(radom\_generation(20, 40), end = " ")

Output: 40 39 34

1. Write a function that takes two numbers and returns the number that has minimum one’s digit.

def check\_for\_ones(num1, num2):

print("The two numbers are " + str(num1) + " and " + str(num2))

li = [num1, num2]

list\_of\_ones = []

basic = "The number with higher one's digit is: "

for i in li:

string\_of\_num = str(i)

last\_val = string\_of\_num[len(string\_of\_num) - 1]

list\_of\_ones.append(last\_val)

if list\_of\_ones[0] > list\_of\_ones [1]:

return basic + str(num1)

else:

return basic + str(num2)

print(check\_for\_ones(10248, 28391))

Ouput:

The two numbers are 10248 and 28391

The number with higher one's digit is: 10248

1. Write a function to check whether the number entered by user is Armstrong number or not

def armstrong(num):

addition = 0

a = num

while num > 0:

dg = num % 10

addition += dg\*\*3

num = num//10

if addition == a:

return f"{a} is an armstrong number."

else:

return f"{a} is not an armstrong number."

print(armstrong(153))

print(armstrong(406))

Output:

153 is an armstrong number.

406 is not an armstrong number.

1. Write a function to generate Fibonacci series up to nth term.

def fib(n):

n1 = 0

n2 = 1

fib\_list = []

for i in range(2, n):

n3 = n1 + n2

n1 = n2

n2 = n3

fib\_list.append(n3)

return fib\_list

print(fib(10))

Output: [1, 2, 3, 5, 8, 13, 21, 34]

1. Write a function to take two strings and check if they are of same length.

def check\_length(string1, string2):

if len(string1) == len(string2):

return "The strings are of the same length."

else:

return "The strings are NOT of the same length."

print(check\_length("Test", "Code"))

print(check\_length("Program", "Test"))

Output:

The strings are of the same length.

The strings are NOT of the same length.

1. Write a program to perform Insertion Sort

def insertion\_sort(list1):

for i in range(len(list1) - 1):

minpos = i

for j in range(i, len(list1)):

if list1[j] < list1[minpos]:

minpos = j

temp = list1[i]

list1[i] = list1[minpos]

list1[minpos] = temp

return list1

print(insertion\_sort([1, 2, 3, 3, 6, 6, 7]))

Output: [1, 2, 3, 3, 6, 6, 7]

1. Write a function to perform Bubble Sort

def bubbleSort(list1):

n = len(list1)

for i in range(n):

for j in range(0, n-i-1):

if list1[j] > list1[j+1] :

list1[j], list1[j+1] = list1[j+1], list1[j]

return list1

print(bubbleSort([2, 6, 2, 4, 6, 1, 7, 7, 2, 3, 9]))

Output: [1, 2, 2, 2, 3, 4, 6, 6, 7, 7, 9]

1. Write a function to perform Binary Search

def binary\_search(list1, low, high, x):

high = len(list1)-1

if high >= low:

mid = (high + low) // 2

if list1[mid] == x:

return mid

elif list1[mid] > x:

return binary\_search(list1, low, mid - 1, x)

else:

return binary\_search(list1, mid + 1, high, x)

else:

return -1

print(binary\_search([0, 2, 4, 5, 2, 4], 0, 0, 4))

Output: 2

1. Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).

import random

print(random.randint(1, 6)

Output: 3

1. Write a program Read a file line by line and print it.

fh = open("poem.txt", "r")

a = fh.readlines()

print(a)

fh.close()

Output:

Humpty Dumpty sat on a Wall,

Humpty Dumty had a great Fall.

1. Write a python program to read a file and print the contents after removing all punctuation marks

fh = open("Test subject.txt", "r")

lines = fh.read()

fh.close()

punctuations = [".", ",", "?", "!", ":"]

string\_var = ""

for i in lines:

if i in punctuations:

pass

else:

string\_var = string\_var + i

print(string\_var)

Output:

Humpty Dumpty sat on a Wall

Humpty Dumpty had a great Fall

1. Write a python program to generate random numbers for password

from random import randint

print(randint(0, 20))

Output: 14

1. Write a program to read the content from a file and write in another file.

file\_1 = open("Test Subject.txt", "r")

file\_1\_data = file\_1.read()

file\_1.close()

file\_2 = open("New File.txt", "w")

file\_2.write(file\_1\_data)

file\_2.close()

1. Write a program to count the words “to” and “the” present in a text file “Article.txt”

file = open("Article.txt", "r")

file\_data = file.read()

file.close()

words = file\_data.split(" ")

count\_to = 0

count\_the = 0

for i in words:

if i == "to":

count\_to += 1

elif i == "the":

count\_the += 1

print(f'Number of "The"s: {count\_the} \n Number of "To"s: {count\_to}')

Output: Number of "The"s: 6

Number of "To"s: 5

1. Write a program that counts uppercase characters present in text file “Poem.txt”.

file = open("Poem.txt", "r")

file\_data = file.read()

file.close()

print(file\_data)

count = 0

for i in file\_data:

if i.isupper():

count += 1

print(count)

Output: 6

1. Write a program to remove all the lines that contain the character `a' in a file and write it to another file.

file\_1 = open("Test Subject.txt", "r")

file\_data = file\_1.readlines()

file\_1.close()

for i in file\_data:

if "a" in i:

file\_data.remove(i)

file\_2 = open("New File.txt", "w")

file\_2.writelines(file\_data)

file\_2.close()

1. Write a program to read a text file and show the longest line from it.

file = open("Test Subject.txt", "r")

file\_data = file.readlines()

file.close()

print(max(file\_data))

Output: a paragraph is a sentence or a group of sentences that supports one central, unified idea.

1. Write a program to copy the lines which are starting from uppercase character to another file. Take the two filenames from user.

file\_name\_1 = input("Enter name of file to be read: ")

file\_name\_2 = input("Enter name of file to write: ")

file1 = open(file\_name\_1 + ".txt", "r")

file1\_data = file1.readlines()

file1.close()

to\_be\_written = []

for i in file1\_data:

if i[0].isupper():

to\_be\_written.append(i)

file2 = open(file\_name\_2 + ".txt", "w")

file2.writelines(to\_be\_written)

file2.close()

1. Write a program to Recursively find the factorial of a natural number.

def fact(n):

if n == 1:

return 1

else:

return n\*fact(n-1)

print(fact(6))

Output: 720

1. Write a recursive code to find the sum of all elements of a list.

def sum\_of\_list(li, n):

if len(li) == 1:

return li[0]

else:

return li[0] + sum\_of\_list(li[1:], n)

list\_1 = [1, 2, 3, 4, 5, 6]

print(sum\_of\_list(list\_1, len(list\_1)))

Output: 21

1. Write a recursive code to compute the nth Fibonacci number.

def Fibonacci(n):

if n<0:

print("Invalid input")

elif n==0:

return 0

elif n==1:

return 1

else:

return Fibonacci(n-1)+Fibonacci(n-2)

print(Fibonacci(8))

Output: 21

1. Write a program to perform recursive binary search.

def binarySearch (arr, l, r, x):

if r >= l:

mid = l + (r - l) // 2

if arr[mid] == x:

return mid

elif arr[mid] > x:

return binarySearch(arr, l, mid-1, x)

else:

return binarySearch(arr, mid + 1, r, x)

else:

return -1

arr = [ 2, 3, 4, 10, 40 ]

x = 10

result = binarySearch(arr, 0, len(arr)-1, x)

if result != -1:

print ("Element is present at index % d" % result)

else:

print ("Element is not present in array")

Output: Element is present at index 3

1. Write a recursive Python program to test if a string is a palindrome or not.

def isPalRec(st, s, e) :

if len(st) == 0:

return True

if (s == e):

return True

if (st[s] != st[e]) :

return False

if (s < e + 1) :

return isPalRec(st, s + 1, e - 1);

return True

strng1 = "rac"

strng2 = "racecar"

print(isPalRec(strng1, 0, len(strng1) -1))

print(isPalRec(strng2, 0, len(strng2) -1))

Output: Flase

True

1. Write a recursive code to compute Greatest Common divisor of two numbers.

def gcd(a,b):

if (a == 0):

return b

if (b == 0):

return a

if (a == b):

return a

if (a > b):

return gcd(a-b, b)

return gcd(a, b-a)

print(gcd(28, 35))

Output: 7

1. Write a recursive code to generate Fibonacci series.

def fib(n):

if n <= 1:

return n

else:

return(fib(n-1) + fib(n-2))

for i in range(0, 10):

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

Output: 0 1 1 2 3 5 8 13 21 34

1. Write a recursive function to reverse a string and print it.

def reverse(string):

if len(string) == 0:

return

temp = string[0]

reverse(string[1:])

print(temp, end='')

reverse("Hello")

Output: olleH

1. Using tkinter library create a graphical counter which increases per second and stops when stop button is pressed.

import tkinter as tk

window = tk.Tk()

window.geometry("500x200")

window.title("Counter")

window.configure(bg = "#aaaaaa")

title\_label = tk.Label(text = "Counter", fg = "#404040", bg = "#aaaaaa", font = ("Arial", 20, "bold"))

title\_label.pack()

sec = 1000

counter = 0

def counter\_label(label):

counter = 0

def count():

global counter

counter += 1

label.config(text = str(counter))

label.after(sec, count)

count()

label = tk.Label(window, fg = "#404040", bg = "#aaaaaa", font = ("Arial", 20, "bold"))

label.pack()

counter\_label(label)

button = tk.Button(window, text = "Stop", fg = "#fff", bg = "#401010", font = ("Arial", 20, "bold"), command = window.destroy)

button.pack()

window.mainloop()

Output:



1. Using tkinter create a program that calculates the sum of two value entered by user and display the result in a label.

import tkinter as tk

font\_def = ("Arial", 20, "bold")

window = tk.Tk()

window.title("Sum calc")

window.config(bg = "#aaaaaa")

window.geometry("400x400")

tk.Label(bg = "#aaaaaa").pack()

num1\_lab = tk.Label(window, bg = "#aaa", text = "Enter number 1", font = font\_def).pack()

num1\_ent = tk.Entry(window, text = "Number 1", font = font\_def)

num1\_ent.pack()

num2\_lab = tk.Label(window, bg = "#aaa", text = "Enter number 2", font = font\_def).pack()

num2\_ent = tk.Entry(window, text = "Number 2", font = font\_def)

num2\_ent.pack()

def display(num1, num2):

summ = num1 + num2

lab.configure(text = str(summ), font = font\_def)

tk.Label(bg = "#aaaaaa").pack()

lab = tk.Label(window, bg = "#aaaaaa", fg = "#401010", font = font\_def)

lab.pack()

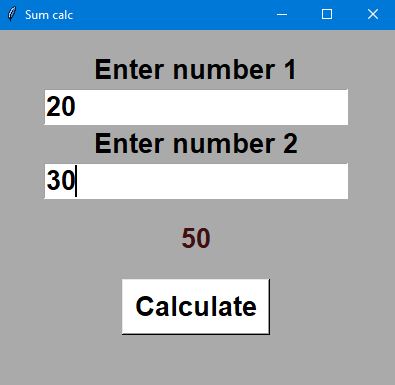
tk.Label(bg = "#aaaaaa").pack()

button = tk.Button(window, text = "Calculate", bg = "#ffffff", font = font\_def, command = display(num1\_ent.get(), num2\_ent.get()))

button.pack()

window.mainloop()

Output:



1. Create a graphical application that accepts user inputs, performs some operation on them, and then writes the output on the screen. For example, write a small calculator. Use the tkinter library.

from tkinter import \*

def click(num):

global operator

operator = operator + str(num)

input\_text.set(operator)

def clear():

global operator

operator = ""

input\_text.set("")

def equal():

global operator

eq = str(eval(operator))

input\_text.set(eq)

window = Tk()

window.title("Calculator")

operator = ""

input\_text = StringVar()

text\_disp = Entry(window, textvariable = input\_text, bd = 10, insertwidth = 3,

bg = "#bbbbbb",fg = "#303030", justify = "right",

font = ("Arial", 60, 'bold')).grid(columnspan=4)

#First row.

bt\_clear = Button(window, bd = 5, fg = "#303030", text = "Clear",

font =("Arial", 30, 'bold'), command = clear

).grid(row = 1, column = 0)

bt\_div = Button(window, bd = 5, fg = "#303030", text = "/", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click("/")

).grid(row = 1, column = 3)

#Second row.

bt7 = Button(window, bd = 5, fg = "#303030", text = 7, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(7)

).grid(row = 2, column = 0)

bt8 = Button(window, bd = 5, fg = "#303030", text = 8, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(8)

).grid(row = 2, column = 1)

bt9 = Button(window, bd = 5, fg = "#303030", text = 9, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(9)

).grid(row = 2, column = 2)

bt\_mult = Button(window, bd = 5, fg = "#303030", text = "\*", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click("\*")

).grid(row = 2, column = 3)

#Third row.

bt4 = Button(window, bd = 5, fg = "#303030", text = 4, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(4)

).grid(row = 3, column = 0)

bt5 = Button(window, bd = 5, fg = "#303030", text = 5, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(5)

).grid(row = 3, column = 1)

bt6 = Button(window, bd = 5, fg = "#303030", text = 6, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(6)

).grid(row = 3, column = 2)

bt\_add = Button(window, bd = 5, fg = "#303030", text = "+", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click("+")

).grid(row = 3, column = 3)

#Third row.

bt1 = Button(window, bd = 5, fg = "#303030", text = 1, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(1)

).grid(row = 4, column = 0)

bt2 = Button(window, bd = 5, fg = "#303030", text = 2, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(2)

).grid(row = 4, column = 1)

bt3 = Button(window, bd = 5, fg = "#303030", text = 3, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(3)

).grid(row = 4, column = 2)

bt\_sub = Button(window, bd = 5, fg = "#303030", text = "-", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click("-")

).grid(row = 4, column = 3)

#Fourth row.

bt\_dec = Button(window, bd = 5, fg = "#303030", text = ".", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(".")

).grid(row = 5, column = 0)

bt0 = Button(window, bd = 5, fg = "#303030", text = 0, padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click(0)

).grid(row = 5, column = 1)

bt\_perc = Button(window, bd = 5, fg = "#303030", text = "%", padx = 50,

font =("Arial", 30, 'bold'), command = lambda:click("%")

).grid(row = 5, column = 2)

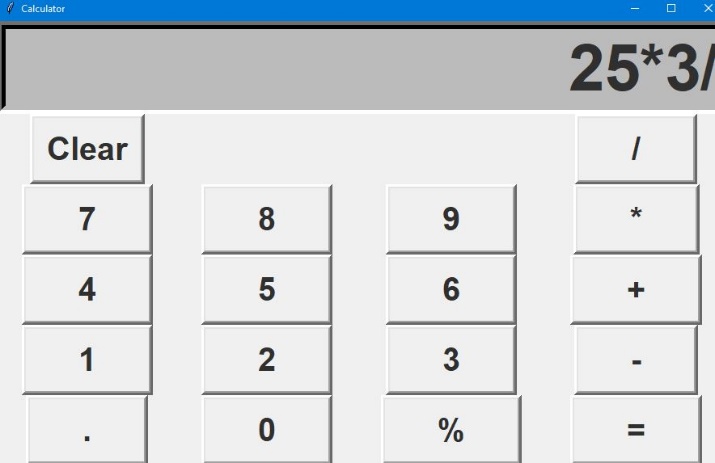
bt\_eq = Button(window, bd = 5, fg = "#303030", text = "=", padx = 50,

font =("Arial", 30, 'bold'), command = equal

).grid(row = 5, column = 3)

window.mainloop()

Output:



1. . Import the math module to calculate factorial, square and log2 of a number.

import math

n = 9

print(math.factorial(n))

print(math.sqrt(n))

print(math.log(n, 2))

Output:

362880

3.0

3.1699250014423126