**Practical File**

**Of**

**Python Programming**

**Using Artificial**

**Intelligence**

**24CAI1101**

***Submitted***

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

COMPUTER SCIENCE & ENGINEERING



**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

December, 2024

**Submitted To: Submitted By:**

Faculty name: Mr. Chirag Jain Name: Harpreet Singh

Designation: Associate Professor RollNo : 2410993279

Chitkara University, Punjab 1st Sem, Batch2024

# INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Practical Name** | **Teacher Sign** |
| 1 | 1. Write a [Python Program to Calculate the Area of a Triangle](https://www.programiz.com/python-programming/examples/area-triangle) 2. Write a [Python Program to Swap Two Variables](https://www.programiz.com/python-programming/examples/swap-variables) 3. Write a [Python Program to Convert Celsius to Fahrenheit](https://www.programiz.com/python-programming/examples/celsius-fahrenheit) |  |
| 2 | 1. Write a [Python Program to Check if a Number is Odd or Even](https://www.programiz.com/python-programming/examples/odd-even) 2. Write a [Python Program to Check if a Number is Positive, Negative or 0](https://www.programiz.com/python-programming/examples/positive-negative-zero) 3. Write a [Python Program to Check Armstrong Number](https://www.programiz.com/python-programming/examples/armstrong-number) |  |
| 3 | 1. Write a Python program to check if a given number is Fibonacci number? 2. Write a Python program to print cube sum of first n natural numbers. 3. Write a Python program to print all odd numbers in a range. |  |
| 4 | 1. Write a Python Program to Print Pascal Triangle 2. WAP to Draw the following Pattern for n number. |  |
| 5 | Write a program with a function that accepts a string from keyboard and create a new string after converting character of each word capitalized. |  |
| 6 | 1. Write a program that accepts a list from user. Your program should reverse the content of list and display it. Do not use reverse () method. 2. Find and display the largest number of a list without using built-in function max (). Your program should ask the user to input values in list from keyboard. |  |
| 7 | Find the sum of each row of matrix of size m x n. |  |
| 8 | a) Write a program that reads a string from keyboard and display:   * The number of uppercase letters in the string. * The number of lowercase letters in the string. * The number of digits in the string. * The number of whitespace characters in the string.  1. [Python Program to Find Common Characters in Two Strings.](https://www.sanfoundry.com/python-program-check-common-letters-string/) 2. [Python Program to Count the Number of Vowels in a String.](https://www.sanfoundry.com/python-program-count-number-vowels-string/) |  |
| 9 | Write a Python program to check if a specified element presents in a tuple of tuples. |  |
| 10 | Write a Program in Python to Find the Differences Between Two Lists Using Sets. |  |
| 11 | 1. Write a Python program Remove duplicate values across Dictionary Values. 2. Write a Python program to Count the frequencies in a list using dictionary in Python. |  |
| 12 | a) Write a Python Program to Capitalize First Letter of Each Word in a File.  b.) Write a Python Program to Print the Contents of File in Reverse Order |  |
| 13 | WAP to catch an exception and handle it using try and except code blocks. |  |
| 14 | Write a Python Program to Append, Delete and Display Elements of a List using Classes. |  |
| 15 | Write a [Python Program to Find the Area and Perimeter of the Circle using Class](https://www.sanfoundry.com/python-program-class-compute-area-perimeter-circle/) |  |
| 16 | Create an interactive application using Python's Tkinter library for graphics programming. |  |

**Program 1:**

**Title of program:** Write a [Python Program to Calculate the Area of a Triangle](https://www.programiz.com/python-programming/examples/area-triangle)

**Objective:** This program will calculate the area of triangle when we will give measure of side.

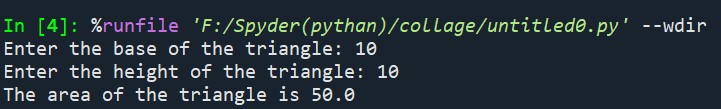
**Code:** def calculate\_triangle\_area(base, height):

return 0.5 \* base \* height base = float(input("Enter the base of the triangle: ")) height = float(input("Enter the height of the triangle: ")) area = calculate\_triangle\_area(base, height) print("The area of the triangle is",area)

**Input:** Base : 10

Height : 10

**Result:**



**Title of program:** Write a [Python Program to Swap Two Variables](https://www.programiz.com/python-programming/examples/swap-variables)

**Objective:** This program will swap the 2 variables entered by us.

**Code:** a=int(input("Enter a number:")) b=int(input("Enter a number:"))

t=a

y=b

print("After swapping")

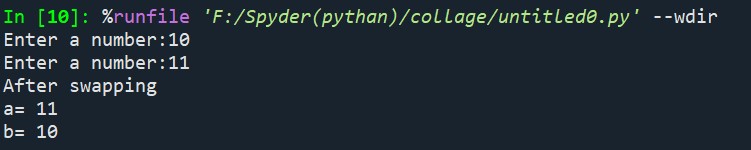
print("a=",y) print("b=",t)

**Input:** a (first variable) = 10

b (second variable) = 11

**Result**

**:**



**Title of program:** Write a [Python Program to Convert Celsius to Fahrenheit.](https://www.programiz.com/python-programming/examples/celsius-fahrenheit)

**Objective:** This program will help us to convert Celsius into Fahrenheit.

**Code:** def celsius\_to\_fahrenheit(celsius):

return (celsius \* 9/5) + 32

celsius = float(input("Enter temperature in Celsius: ")) fahrenheit = celsius\_to\_fahrenheit(celsius) print(celsius,"°C is equal to",fahrenheit,"°F")

**Input:**  Celsius = 10

**Result:**



**Program 2:**

**Title of program:** Write a [Python Program to Check if a Number is Odd or Even.](https://www.programiz.com/python-programming/examples/odd-even)

**Objective:** This program will help us to check weather a number is odd or even .

**Code:** def check\_odd\_even(number): if number % 2 == 0:

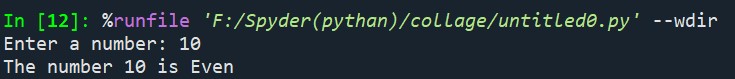
return "Even" else:

return "Odd"

number = int(input("Enter a number: ")) result = check\_odd\_even(number) print("The number",number,"is",result)

**Input:** Number to be checked = 10

**Result:**



**Title of program:** Write a [Python Program to Check if a Number is Positive, Negative or](https://www.programiz.com/python-programming/examples/positive-negative-zero)

[0.](https://www.programiz.com/python-programming/examples/positive-negative-zero)

**Objective:** This program helps us to check the number is positive, negative or zero.

**Code:** def check\_number(number): if number > 0: return "Positive" elif number < 0: return "Negative" else:

return "Zero"

number = float(input("Enter a number: ")) result = check\_number(number) print("The number","is",result)

**Input:** Enter number = 10

**Result:**



**Title of program:** Write a [Python Program to Check Armstrong Number.](https://www.programiz.com/python-programming/examples/armstrong-number)

**Objective:** This program checks whether the number is Armstrong number or not.

**Code:**  def is\_armstrong(number):

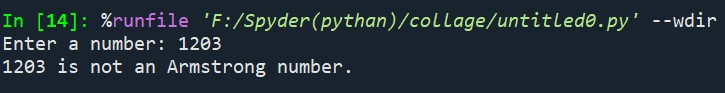
digits = str(number) num\_digits = len(digits)

sum\_of\_powers = sum(int(digit) \*\* num\_digits for digit in digits) return sum\_of\_powers == number number = int(input("Enter a number: ")) if is\_armstrong(number): print(number,"is an Armstrong number.") else:

print(number,"is not an Armstrong number.")

**Input:** Number = 1203

**Result:**



**Program 3:**

**Title of program:** Write a Python program to check if a given number is Fibonacci number?

**Objective:** This program checks whether the number is Fibonacci or not.

**Code:** import math

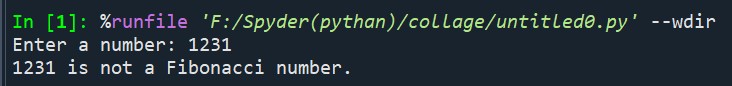
def is\_perfect\_square(x): s = int(math.sqrt(x)) return s \* s == x def is\_fibonacci\_number(n): return is\_perfect\_square(5 \* n \* n + 4) or is\_perfect\_square(5 \* n \* n - 4) number = int(input("Enter a number: ")) if is\_fibonacci\_number(number):

print(number, "is a Fibonacci number.") else:

print(number,"is not a Fibonacci number.")

**Input:** Number =1231

**Result:**



**Title of program:** Write a Python program to print cube sum of first n natural numbers.

**Objective:** This program helps us to find cube sum of first n natural number.

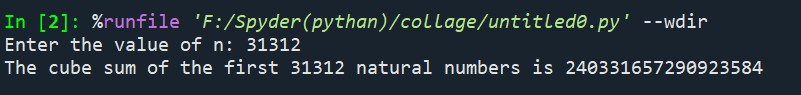
**Code:** def cube\_sum(n):

return sum(i\*\*3 for i in range(1, n+1)) n = int(input("Enter the value of n: ")) result = cube\_sum(n)

print("The cube sum of the first",n,"natural numbers is",result)

**Input:** value of n = 31312

**Result:**



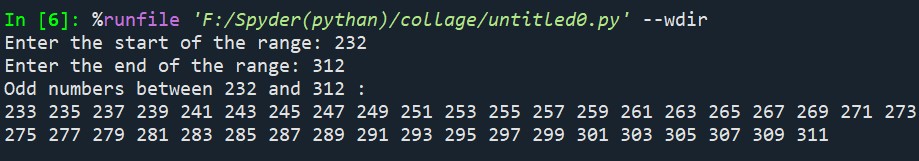
**Title of program:** Write a Python program to print all odd numbers in a range.

**Objective:** This program helps print all odd numbers in range.

**Code:** def print\_odd\_numbers(start, end): for number in range(start, end + 1): if number % 2 != 0: print(number, end=' ') start = int(input("Enter the start of the range: ")) end = int(input("Enter the end of the range: ")) print("Odd numbers between",start,"and",end,":") print\_odd\_numbers(start, end)

**Code:** Range = 232

**Result:**



**Program 4:**

**Title of program:** Write a Python Program to Print Pascal Triangle.

**Objective:** This program helps us to print Pascal Triangle.

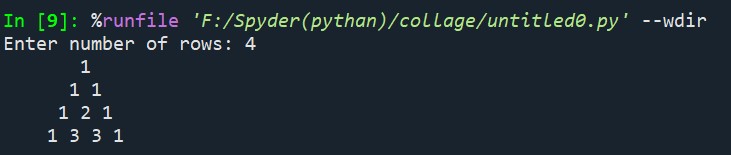
**Code:** def print\_pascals\_triangle(n):

triangle = [] for row in range(n):

new\_row = [1] \* (row + 1) for col in range(1, row): new\_row[col] = triangle[row - 1][col - 1] + triangle[row - 1][col] triangle.append(new\_row) width = n \* 2 for row in triangle: print(" " \* (width - len(row)), end="") print(" ".join(map(str, row))) n = int(input("Enter number of rows: ")) print\_pascals\_triangle(n)

**Input:** number of row = 4

**Output:**



**Title of program:** WAP to draw the following Pattern for n number:

1. 1 1 1 1
2. 2 2 2
3. 3 3
4. 4

5

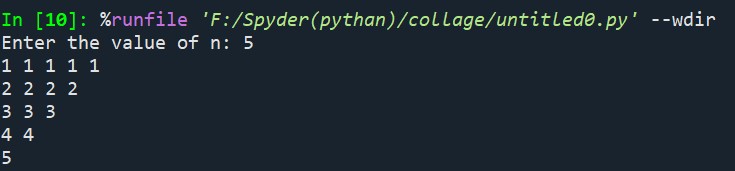
**Objective:** This program will draw the pattern above for n numbers.

**Code:** def draw\_pattern(n): for i in range(1, n + 1): for j in range(n - i + 1): print(i, end=' ') print()

n = int(input("Enter the value of n: ")) draw\_pattern(n)

**Input:** N = 5

**Output:**



**Program 5:**

**Title of program:** Write a program with a function that accepts a string from keyboard and create a new string after converting character of each word capitalized.

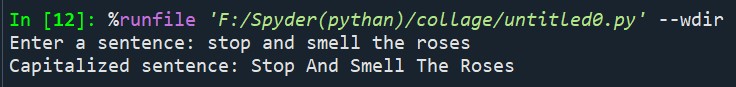
**Objective:** This program will help us to capitalized all the words in a sentence.

**Code:** def capitalize\_words(sentence):

capitalized\_sentence = sentence.title() return capitalized\_sentence sentence = input("Enter a sentence: ") result = capitalize\_words(sentence) print("Capitalized sentence:", result)

**Input:** Sentence =stop and smell the roses

**Result:**



**Program 6:**

**Title of program:** Write a program that accepts a list from user. Your program should reverse the content of list and display it. Do not use reverse () method.

**Objective:** This program helps us to reverse the list.

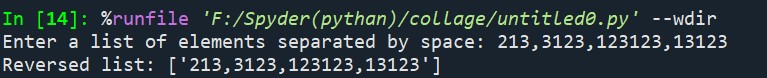
**Code:** def reverse\_list(lst):

reversed\_lst = [] for i in range(len(lst) - 1, -1, -1): reversed\_lst.append(lst[i]) return reversed\_lst

user\_input = input("Enter a list of elements separated by space: ") lst = user\_input.split() reversed\_lst = reverse\_list(lst) print("Reversed list:", reversed\_lst)

**Input:** list = 213,3123,123123,13123

**Result:**



**Title of program:** Find and display the largest number of a list without using built-in function max (). Your program should ask the user to input values in list from keyboard.

**Objective:** This program will display largest number in the list.

**Code:** def find\_largest\_number(lst):

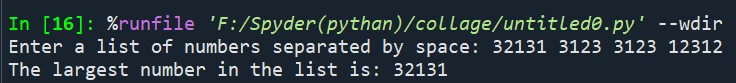
largest = lst[0] for number in lst: if number > largest: largest = number return largest

user\_input = input("Enter a list of numbers separated by space: ") lst = list(map(int, user\_input.split())) largest\_number = find\_largest\_number(lst)

print("The largest number in the list is:", largest\_number)

**Input:** List = 32131 3123 3123 12312

**Output:**



**Program 7:**

**Title of program:** Find the sum of each row of matrix of size m x n.

**Objective:** This program will find sum of each row in a matrix of size m x n.

**Code:** def sum\_of\_rows(matrix):

row\_sums = [] for row in matrix: row\_sums.append(sum(row))

return row\_sums m = int(input("Enter the number of rows (m): ")) n = int(input("Enter the number of columns (n): ")) matrix = [] print("Enter the matrix elements row-wise:") for i in range(m): row = list(map(int, input().split())) matrix.append(row) row\_sums = sum\_of\_rows(matrix) for i in range(m):

print(f"Sum of row {i + 1} = {row\_sums[i]}")

**Input:** Number of rows = 3

Number of columns = 4

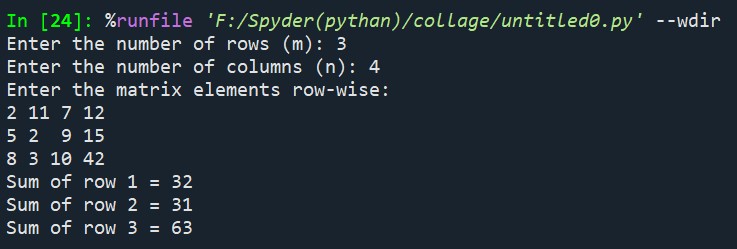
Matrix = 2 11 7 12

5 2 9 15

8 3 10 42

**Result**

**:**



**Program 8:**

**Title of program:** Write a program that reads a string from keyboard and display:

* The number of uppercase letters in the string.
* The number of lowercase letters in the string.
* The number of digits in the string.
* The number of whitespace characters in the string.

**Objective:** This program will tell us letters are in uppercase or lowercase in string or there how many digits and backspace are there.

**Code:**  def count\_characters(s):

uppercase\_count = 0 lowercase\_count = 0 digit\_count = 0 whitespace\_count = 0 for char in s: if char.isupper():

uppercase\_count += 1 elif char.islower():

lowercase\_count += 1 elif char.isdigit():

digit\_count += 1 elif char.isspace():

whitespace\_count += 1

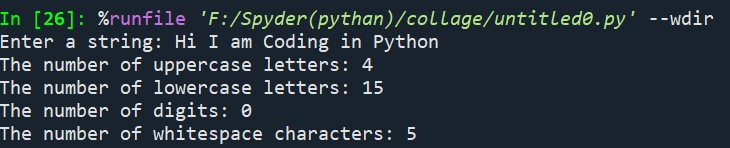
return uppercase\_count, lowercase\_count, digit\_count, whitespace\_count input\_string = input("Enter a string: ") uppercase\_count, lowercase\_count, digit\_count, whitespace\_count = count\_characters(input\_string) print("The number of uppercase letters:",uppercase\_count) print(f"The number of lowercase letters:",lowercase\_count) print(f"The number of digits:",digit\_count)

print(f"The number of whitespace characters:",whitespace\_count)

**Input:** The line to be checked = Hi I am Coding in Python

**Result**

**:**



**Title of program:** [Python Program to Find Common Characters in Two Strings.](https://www.sanfoundry.com/python-program-check-common-letters-string/)

**Objective:** This program will help to find common characters in two strings.

**Code:**  def find\_common\_characters(str1, str2):

set1 = set(str1) set2 = set(str2)

common\_chars = set1.intersection(set2) return ''.join(sorted(common\_chars)) string1 = input("Enter the first string: ") string2 = input("Enter the second string: ")

common\_characters = find\_common\_characters(string1, string2) print("Common characters:",common\_characters)

**Input:** First string = Aadit

Second string = Aadit Gupta

**Result**

**:**



**Title of program:**  [Python Program to Count the Number of Vowels in a String.](https://www.sanfoundry.com/python-program-count-number-vowels-string/)

**Objective:** This program will count the number of vowels in a string.

**Code:** def count\_vowels(s):

vowels = "aeiouAEIOU"

count = 0 for char in s: if char in vowels:

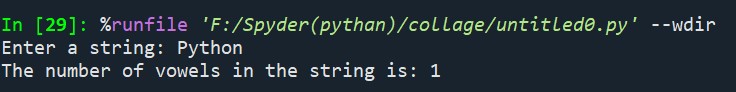
count += 1 return count

input\_string = input("Enter a string: ") vowel\_count = count\_vowels(input\_string)

print("The number of vowels in the string is:",vowel\_count)

**Input:** string = Python

**Result:**



**Program 9:**

**Title of program:** Write a Python program to check if a specified element presents in a tuple of tuples. Original list:((‘Red’ ,’White’ , ‘Blue’),(‘Green’, ’Pink’ , ‘Purple’), (‘Orange’,

‘Yellow’, ‘Lime’)) Check if White present in said tuple of tuples! True Check if Olive present in said tuple of tuples! False

**Objective:** This programwill check whether the mentioned elements are present in the tuple or not .

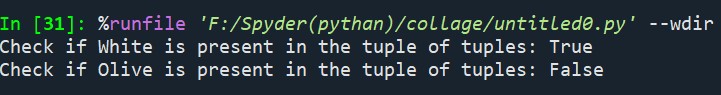
**Code:** def check\_element\_in\_tuples(tuples, element): for sub\_tuple in tuples: if element in sub\_tuple:

return True return False

tuples = (('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange', 'Yellow', 'Lime')) element1 = 'White' element2 = 'Olive'

is\_element1\_present = check\_element\_in\_tuples(tuples, element1) is\_element2\_present = check\_element\_in\_tuples(tuples, element2) print("Check if",element1,"is present in the tuple of tuples:",is\_element1\_present) print("Check if",element2,"is present in the tuple of tuples:",is\_element2\_present)

**Result:**



**Title of program:** Write a Python program to remove an empty tuple(s) from a list of tuples.

**Objective:** This program will remove the empty space in tuples.

**Code:** def remove\_empty\_tuples(lst):

return [t for t in lst if t]

sample\_data = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')] result = remove\_empty\_tuples(sample\_data) print("After removing empty tuples:", result)

**Result:**



**Program 10:**

**Title of program:** Write a Program in Python to Find the Differences between Two Lists Using Sets.

**Objective:** This program will show the differences in two sets.

**Code:** def find\_differences(list1, list2):

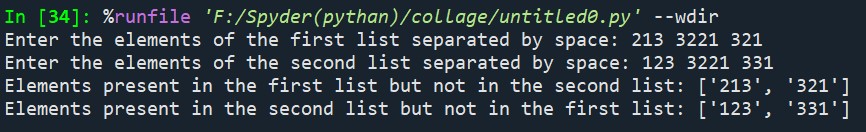
set1 = set(list1) set2 = set(list2) diff1 = set1 - set2 diff2 = set2 - set1 return list(diff1), list(diff2)

list1 = input("Enter the elements of the first list separated by space: ").split() list2 = input("Enter the elements of the second list separated by space: ").split() differences1, differences2 = find\_differences(list1, list2) print("Elements present in the first list but not in the second list:",differences1) print(f"Elements present in the second list but not in the first list:",differences2)

**Input:** Elements of first list = 213 3221 321

Elements of second list = 123 3221 331

**Output:**



**Program 11:**

**Title of program:** Write a Python program Remove duplicate values across Dictionary Values.

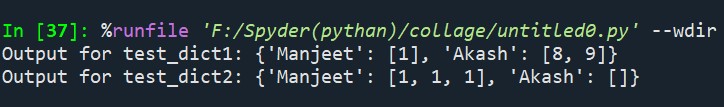
**Objective:** This program will remove duplicate vales in dictionary.

**Code:** def remove\_duplicates(test\_dict):

seen\_values = set() for key in test\_dict:

new\_list = [val for val in test\_dict[key] if val not in seen\_values] seen\_values.update(test\_dict[key]) test\_dict[key] = new\_list return test\_dict test\_dict1 = {'Manjeet': [1], 'Akash': [1, 8, 9]} test\_dict2 = {'Manjeet': [1, 1, 1], 'Akash': [1, 1, 1]} result1 = remove\_duplicates(test\_dict1) result2 = remove\_duplicates(test\_dict2) print("Output for test\_dict1:", result1) print("Output for test\_dict2:", result2)

**Output:**



**Title of program:** Write a Python program to Count the frequencies in a list using dictionary in Python.

**Objective:** This program will count the frequencies in a list.

**Code:** def count\_frequencies(lst): frequency\_dict = {} for item in lst: if item in frequency\_dict:

frequency\_dict[item] += 1 else:

frequency\_dict[item] = 1 return frequency\_dict

input\_list = [1, 1, 1, 5, 5, 3, 1, 3, 3, 1, 4, 4, 4, 2, 2, 2, 2]

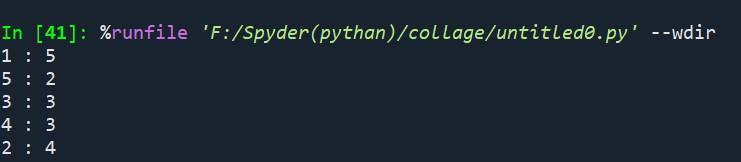
frequencies = count\_frequencies(input\_list) for key, value in frequencies.items():

print(key,":",value)

**Input:** Input =[1, 1, 1, 5, 5, 3, 1, 3, 3, 1,4, 4, 4, 2, 2, 2, 2]

**Result**

**:**



**Program 12:**

**Title of program:** Write a Python Program to Capitalize First Letter of Each Word in a

File.

**Objective:** This program will Capitalize first letter of each word in a file.

**Code:** with open("TestCase.txt",'r') as f: for line in f:

l=line.title() print(l)

**Result:**



**Title of program:** Write a Python Program to Print the Contents of File in Reverse Order.

**Objective:** This program will print the contents of file in reverse Order.

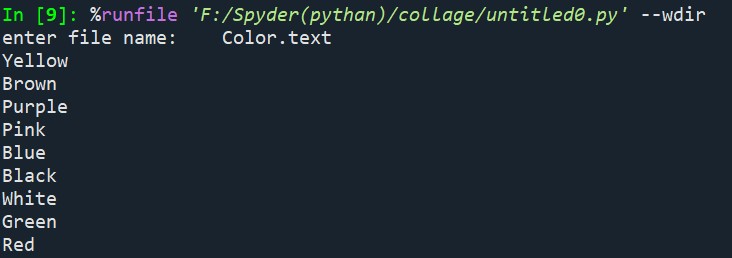
**Solution:** filename = input("Enter the file name : ")

for line in reversed(list(open(filename))):

print(line.rstrip())

**Input:** File name= Color.text

**Output:**



**Program 13:**

**Title of program:** WAP to catch an exception and handle it using try and except code blocks.

**Objective:** This code will give the dividend after dividing numerator and denominator.

**Code:** def divide\_numbers(a, b): try: result = a / b print(f"The result of {a} divided by {b} is {result}") except ZeroDivisionError: print("Error: Division by zero is not allowed.") except TypeError: print("Error: Both inputs must be numbers.") except Exception as e: print(f"An unexpected error occurred: {e}") a = input("Enter the numerator: ")

b = input("Enter the denominator: ") try:

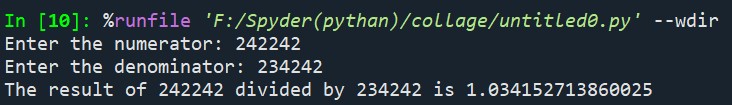
a = int(a) b = int(b) except ValueError: print("Error: Both inputs must be valid integers.") else:

divide\_numbers(a, b)

**Input:** Numerator : 242242

Denominator: 234242

**Result:**



**Program 14:**

**Title of program:** Write a Python Program to Append, Delete and Display Elements of a List using Classes.

**Objective:** This program will help us to append, delete and display elements of a list.

**Code:** class check(): def \_\_init\_\_(self): self.n=[] def add(self,a):

return self.n.append(a) def remove(self,b): self.n.remove(b) def dis(self): return (self.n) obj = check() choice = 1 while choice != 0: print("0.Exit") print("1.Add") print("2.Delete")

print("3.Display") choice = int(input("ENTER CHOICE : ")) if choice == 1:

n = int(input("ENTER THE NUMBER TO APPEND : ")) obj.add(n) print("List : ",obj.dis()) elif choice == 2:

n = int(input("ENTER THE NUMBER TO REMOVE : ")) obj.remove(n) print("List : ",obj.dis()) elif choice == 3:

print("List : ",obj.dis()) elif choice == 0: print("Exiting") else: print("Invalid Choice ! ") print()

**Input:** Enter choice: 0

Enter the number to append: 2

Enter choice:3

Enter choice:0

**Result:**



**Program 15:**

**Title of program:** Write a [Python Program to Find the Area and Perimeter of the Circle using Class.](https://www.sanfoundry.com/python-program-class-compute-area-perimeter-circle/)

**Objective:** This program will give us the area and perimeter of circle when we will give the radius of circle.

**Code:** import math

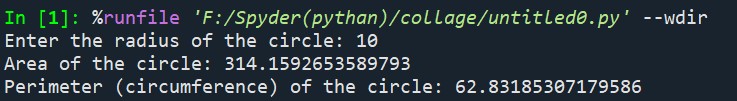
class Circle: def \_\_init\_\_(self, radius): self.radius = radius def area(self): return math.pi \* self.radius \*\* 2 def perimeter(self): return 2 \* math.pi \* self.radius

radius = float(input("Enter the radius of the circle: ")) circle = Circle(radius) circle\_area = circle.area() circle\_perimeter = circle.perimeter() print("Area of the circle:",circle\_area)

print("Perimeter (circumference) of the circle:",circle\_perimeter)

**Input:** Radius of Circle = 10

**Result:**



**Program 16:**

**Title of program:** Create an interactive application using Python's Tkinter library for graphics programming.

**Objective:** A working calculator.

**Code:** from tkinter import \*

root = Tk()

root.title("Simple Calculator")

entry\_field = Entry(root, width=35, borderwidth=5)

entry\_field.grid(row=0, column=0, columnspan=4) def button\_click(number): current\_value = entry\_field.get() entry\_field.delete(0, END) entry\_field.insert(0, str(current\_value) + str(number)) def operator\_click(operator): current\_value = entry\_field.get() entry\_field.delete(0, END) entry\_field.insert(0, str(current\_value) + str(operator)) def clear\_button\_click():

entry\_field.delete(0, END) def equals\_button\_click(): try:

result = eval(entry\_field.get()) entry\_field.delete(0, END) entry\_field.insert(0, str(result)) except Exception as e: entry\_field.delete(0, END) entry\_field.insert(0, "Error")

button\_1 = Button(root, text="1", padx=40, pady=20, command=lambda: button\_click(1)) button\_2 = Button(root, text="2", padx=40, pady=20, command=lambda: button\_click(2)) button\_3 = Button(root, text="3", padx=40, pady=20, command=lambda: button\_click(3)) button\_4 = Button(root, text="4", padx=40, pady=20, command=lambda: button\_click(4)) button\_5 = Button(root, text="5", padx=40, pady=20, command=lambda: button\_click(5)) button\_6 = Button(root, text="6", padx=40, pady=20, command=lambda: button\_click(6)) button\_7 = Button(root, text="7", padx=40, pady=20, command=lambda: button\_click(7)) button\_8 = Button(root, text="8", padx=40, pady=20, command=lambda: button\_click(8)) button\_9 = Button(root, text="9", padx=40, pady=20, command=lambda: button\_click(9)) button\_0 = Button(root, text="0", padx=40, pady=20, command=lambda: button\_click(0)) button\_add = Button(root, text="+", padx=39, pady=20, command=lambda:

operator\_click("+"))

button\_subtract = Button(root, text="-", padx=40, pady=20, command=lambda:

operator\_click("-")) button\_multiply = Button(root, text="\*", padx=40, pady=20, command=lambda:

operator\_click("\*")) button\_divide = Button(root, text="/", padx=40, pady=20, command=lambda:

operator\_click("/")) button\_equals = Button(root, text="=", padx=91, pady=20, command=equals\_button\_click) button\_clear = Button(root, text="Clear", padx=79, pady=20, command=clear\_button\_click) button\_1.grid(row=3, column=0) button\_2.grid(row=3, column=1) button\_3.grid(row=3, column=2) button\_4.grid(row=2, column=0) button\_5.grid(row=2, column=1) button\_6.grid(row=2, column=2) button\_7.grid(row=1, column=0) button\_8.grid(row=1, column=1) button\_9.grid(row=1, column=2) button\_0.grid(row=4, column=0) button\_add.grid(row=1, column=3) button\_subtract.grid(row=2, column=3) button\_multiply.grid(row=3, column=3) button\_divide.grid(row=4, column=3) button\_equals.grid(row=4, column=1, columnspan=2) button\_clear.grid(row=5, column=0, columnspan=4) root.mainloop()

**Input:** calculation to be performed=100\*21

**Result:**

