**1. Python program to check if the input number is prime or not**

# take input from the user

num = int(input("Enter a number: "))

# prime numbers are greater than 1

if num > 1:

# check for factors

for i in range(2,num):

if (num % i) == 0:

print(num,"is not a prime number")

print(i,"times",num//i,"is",num)

break

else:

print(num,"is a prime number")

# if input number is less than or equal to 1, it is not prime

else:

print(num,"is not a prime number")

**2. Program make a simple calculator that can add, subtract, multiply and divide using functions**

# This function adds two numbers

def add(x, y):

return x + y

# This function subtracts two numbers

def subtract(x, y):

return x - y

# This function multiplies two numbers

def multiply(x, y):

return x \* y

# This function divides two numbers

def divide(x, y):

return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

# Take input from the user

choice = input("Enter choice(1/2/3/4):")

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

if choice == '1':

print(num1,"+",num2,"=", add(num1,num2))

elif choice == '2':

print(num1,"-",num2,"=", subtract(num1,num2))

elif choice == '3':

print(num1,"\*",num2,"=", multiply(num1,num2))

elif choice == '4':

print(num1,"/",num2,"=", divide(num1,num2))

else:

print("Invalid input")

**3.Python program to find the factorial of a number provided by the user.**

# take input from the user

num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

for i in range(1,num + 1):

factorial = factorial\*i

print("The factorial of",num,"is",factorial)

**5. Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:**

Score Grade

>= 0.9 A

>= 0.8 B

>= 0.7 C

>= 0.6 D

< 0.6 F

Run the program repeatedly as shown above to test the various, different values for input.

input1 = input('Enter score: ')

try:

score = float(input1) #Only allows input floats

except:

print('Bad score')

quit()

if score > 0 and score < 1: #Scores must be between 0.0 and 1.0

if score >= .9:

print('A')

elif score >= .8:

print('B')

elif score >= .7:

print('C')

elif score >= .6:

print('D')

else:

print('F')

else:

print('Bad score')

**6. Rewrite the grade program from previous chapter using a function called computegrade that takes a score as its parameter and returns a grade as a string.**

def computegrade(score):

if score > 0 and score < 1:

if score >= .9:

print('A')

elif score >= .8:

print('B')

elif score >= .7:

print('C')

elif score >= .6:

print('D')

else:

print('F')

else:

print('Bad score')

input\_score = input('Enter score: ')

try:

score = float(input\_score) #Only allows input floats

except:

print('Bad score')

quit()

computegrade(score)

**7. Generating random numbers.Run the program on your system and see what numbers you get. Run the program more than once and see what numbers you get.**

import random

for i in range(10):

x = random.random()

print(x)

**8. Write a program which repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.**

count = 0 #Initializes values

total = 0

while True: #Stays in loop until break

input\_number = input('Enter a number: ')

if input\_number == 'done' : break #Exits the loop

try:

number = float(input\_number) #Only allows input floats

except:

print('Invalid input')

count = count + 1 #Counter

total = total + number #Running total

average = total / count #Computes the average

**9. Write a program to open the file words.txt and read it line by line. For each line, split the line into a list of words using the split function.For each word, check to see if the word is already in a list. If the word is not in the list, add it to the list. When the program completes, sort and print the resulting words in alphabetical order.**

myList = []

fhand = open('words.txt')

for line in fhand:

words = line.split() #Splits line into array of words

for word in words:

if word in myList : continue #Discards duplicates

myList.append(word) #Updates the list

print(sorted(myList)) #Alphabetical order

**10. Write a program that reads the words in words.txt and stores them as keys in a dictionary.Then use the 'in' operator as a fast way to check whether a string is in the dictionary.**

count = 0

dictionary\_words = dict() #Initializes the dictionary

fhand = open('words.txt')

for line in fhand:

words = line.split()

for word in words:

count = count + 1

if word in dictionary\_words : continue #Discards duplicates

dictionary\_words[word] = count #Value is first time word appears

if 'Python' in dictionary\_words:

print('True')

else:

print('False')

**11. Write a program that reads a file and prints the letters in decreasing order of frequency. Your program should convert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, puntuaction, or anything other than letters a-z.**

import string

counts = 0

dictionary\_counts = dict() #Initializes the dictionary

fname = input('Enter file name: ')

try:

fhand = open(fname)

except:

print('File cannot be opened:', fname)

exit()

for line in fhand:

line = line.translate(str.maketrans('', '', string.digits))

line = line.translate(str.maketrans('', '', string.punctuation))

line = line.lower()

#Removes numbers then punctuation, and lower cases the letters

words = line.split()

for word in words:

for letter in word:

counts += 1 #counts each letter for relative frequencies

if letter not in dictionary\_counts:

dictionary\_counts[letter] = 1

else:

dictionary\_counts[letter] += 1

relative\_lst = list() #Initializes the list

for key, val in list(dictionary\_counts.items()):

relative\_lst.append((val/counts,key)) #Computes the relative frequency

relative\_lst.sort(reverse=True) #Sorts from highest rel freq

for key, val in relative\_lst:

print(key,val)