RASHI JAIN, PGID-11920010

In [ ]:
```
#1. Suppose the cover price of a book is Rs. 200, but bookstores get a 25% discou
# Shipping costs Rs. 40 for the first copy and Rs. 10 for each additional copy.
# Write a Python program to calculate the total wholesale cost for 60 copies

cvr_price = 200  #Cover price of book
dis = 0.25       #discount to bookstore
tot_copies = 60  #Total number of copies

dealer_price = cvr_price*(1 - dis)  #Price at which bookstore is getting book
first_shipp = 40                    #shipping cost for first copy of book
rest_shipp = 10                     #Shipping cost for rest copies of book

tot_shipp = first_shipp + ((tot_copies -1) * rest_shipp)            #total shipp

tot_wholesale_cost = (tot_copies * dealer_price) + tot_shipp        #total whole

print("Total wholesale cost", tot_wholesale_cost)
```

In [2]:
```
#2. Write a function in Python that takes parameters x and y and returns True if

def ispower(x,y):
    if(x==1):
        return True

    if(x % y != 0):
        return False

    return ispower(x/y,y)
#funtion ends here

#taking input from user
x = int(input())
y = int(input())

print("Is", x, "a power of",y,"?:", ispower(x,y))
```

```
32
2
Is 32 a power of 2 ?: True
```

In [6]:
```
#3. GCD of three variables

import math

def result(a, b, c):
    temp = math.gcd(a,b)   #finding gcd for two variables first
    return math.gcd(c, temp)
#funtion ends here

#taking input from user
x = int(input())
y = int(input())
z = int(input())

print("GCD of [",x,y,z,"] is:",result(x,y,z))
```

```
12
15
18
GCD of [ 12 15 18 ] is: 3
```

In [9]:
```
#4. Python script that reads the current time and converts it to a time of day i
# hours, minutes, and seconds, plus the number of days since the epoch

import datetime

x = datetime.datetime.now() #Reading current time
print(x)

print(x.strftime("HOUR: %H, MINUTE: %M, SECOND %S"))
print('NUMBER OF DAYS SINCE EPOCH: ',x - datetime.datetime(1970,1,1,0,0))
```

```
2019-09-06 23:26:00.755876
HOUR: 23, MINUTE: 26, SECOND 00
NUMBER OF DAYS SINCE EPOCH:  18145 days, 23:26:00.755876
```

In [1]:
```
#5. Function that takes three integers as arguments, and that prints either "Yes"
# depending on whether you can or cannot form a triangle from the given lengths


#Function to check whether a,b,c can form triangle or not
def isTriangle(a,b,c):
    if( ((a+b)>c) and ((b+c)>a) and ((a+c)>b) ):
       return True
    else:
       return False
#funtion ends here

#taking input from user
a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))

print("Can we form triangle from [",a,b,c,"] ? :",isTriangle(a,b,c))
```

```
Enter a: 7
Enter b: 6
Enter c: 5
Can we form triangle from [ 7 6 5 ] ? : True
```

In [28]:
```
#7. function for cummulative function
def cummulative_sum(t):
    length = len(t)
    for i in range(1, length):
        t[i] += t[i-1]
    return t
#funtion ends here


#Program for cummulative function
t = []
length = int(input("Enter number of elements in list: "))

for i in range(0,length):
    element = int(input("element:"))
    t.append(element)  #adding element in list

print("List is: ",t)
print("cummulative sum is: " , cummulative_sum(t))
```

```
Enter number of elements in list: 3
element:3
element:2
element:1
List is:  [3, 2, 1]
cummulative sum is:  [3, 5, 6]
```

In [ ]:
```

```

In [70]:
```
# 8. Program that reads a file, breaks each line into words, strips
# whitespace and punctuation from the words, and converts them to uppercase

import string

def new_text(filename):
    modified_text = []
    input_file = open(filename)
    pat =string.punctuation+string.whitespace
    ans=''
    translation_table = dict.fromkeys(map(ord, pat), None)
    line = input_file.readline()
    while line:
        line = line.translate(translation_table)
        ans+=line
        line = input_file.readline()

    input_file.close()
    output_file=open(r"result.txt",'w')
    output_file.write(ans.upper())
    output_file.close()
    return ans.upper()


fn = input() #taking filepath from User
print("Stripped file", new_text(fn))
```

```
C:\input.txt
Stripped file KSDFJKLJSFLW90KSLSF098U
```

In [ ]:
```
#9. Program that searches a directory and all of its subdirectories recursively,
# and returns a list of complete paths for all files with a given suffix


import os

def file_recur(path):
    newlist = os.listdir(path)  #Storing all the files and directories in the giv
    anslist = []   #Empty list to store filename along with their path
    for i in newlist:
        if(os.path.isdir(path+'\\'+i)):  #Checking whether a directory or not
            anslist.extend(file_recur(path+'\\'+i))   #Recursive call to check fo
                                                      #Then adding Files present

        if(os.path.isfile(path+'\\'+i)): #If it is a file, add it directly to Ans
            anslist.append(path+'\\'+i)
    #for loop ends here

    return anslist
#End of Function


fn = input() #taking filepath from User
file_recur(new_text(fn))
```