In [ ]:
```
name   :DINESH KUMAR
rollno :225229108
```

In [ ]:
```
```

In [18]:
```
with open("loginfile.txt","w") as dinesh:
    b=int(input("no of users:"))
    for i in range (b):
        c=input("enter user ID:")
        d=input("enter passwrd:")
        dinesh.write(c)
        dinesh.write(d)
    dinesh.close()
```

```
no of users:3
enter user ID:dk
enter passwrd:78
enter user ID:dinesh
enter passwrd:98
enter user ID:kumar
enter passwrd:10
```

In [19]:
```
dk=open("loginfile.txt","r")
section=open("security.txt","w")
w=dk.read()
section.write(w)
section.close()
```

In [20]:
```
section=open("security.txt","r")
print(section.read())
section.close()
```

```
dk78dinesh98kumar10
```

In [21]:
```
section1=open('security.txt',"r")
r=section1.read()
id=input("user ID:")
ps=input("passwrd:")
if id+ps in r:
    print("login successful")
else:
    print("login failed")
section1.close()
```

```
user ID:dk
passwrd:78
login successful
```

In [ ]:
```
```

In [1]:
```
m=open("marks.txt","a")
n=int=int(input("no of marks:"))
l=[]
for i in range(n):
    k=float(input("enter value:"))
    l.append(k)
m.write(str(l))
m.close()
```

```
no of marks:7
enter value:89
enter value:99
enter value:85
enter value:45
enter value:58
enter value:92
enter value:87
```

In [3]:
```
m=open("marks.txt","r")
m.read()
print("top mark:",max(l))
l.sort(reverse=True)
print("top 3 marks:",(l[:3]))
l.sort()
print("low 3 marks:",(l[:3]))
print("low mark:",min(l))
m.close()
```

```
top mark: 99.0
top 3 marks: [99.0, 92.0, 89.0]
low 3 marks: [45.0, 58.0, 85.0]
low mark: 45.0
```

In [ ]:
```

```

In [4]:
```
while True:
    stock_name = str(input("Enter the name: "))
    file = open("stock_price.txt","a")
    file.write(stock_name)
    file.write('\t')
    n = int(input("Enter the no of stocks: "))
    for i in range(n):
        p = input("cost: ")
        file.write(p)
        file.write("\t")
    file.write("\n")
    quit = input("To Quit = q: ")
    if quit =='q':
        break
file.close()
```

```
Enter the name: asus
Enter the no of stocks: 7
cost: 741
cost: 852
cost: 963
cost: 123
cost: 456
cost: 789
cost: 951
To Quit = q: q
```

In [10]:
```
for stock_price in open ("stock_price.txt","r").readlines():
    pr =[]
    c = stock_price.split()
    print(c)
    for i in range (1,5):
        pr.append(int(i))
    print(max(pr))
    print(min(pr))
    s = sum(pr)
    avg = s/5
    print(avg)
    print('\n')
```

```
['iphone', '890', '524', '789']
4
1
2.0


['iphone', 'apple', 'iphone', '741', '852', '963', '789', '456', '123', '951']
4
1
2.0


['iphone', '4', '2', '3', '1', '6', '5', '0']
4
1
2.0


['asus', '741', '852', '963', '123', '456', '789', '951']
4
1
2.0
```

In [12]:
```
for stock_price in open ("stock_price.txt","r").readlines():
    pr =[]
    c = stock_price.split()
    print(c[0])
    for i in range (1,5):
        pr.append(int(i))
    max_price = max(pr)
    min_price = min(pr)
    a = pr.index(max_price)
    b = pr.index(min_price)
    print("Maximum Price of ",max_price,"on the",a+1,"day")
    print("Minimum Price of ",min_price,"on the",b+1,"day")
```

```
iphone
Maximum Price of  4 on the 4 day
Minimum Price of  1 on the 1 day
iphone
Maximum Price of  4 on the 4 day
Minimum Price of  1 on the 1 day
iphone
Maximum Price of  4 on the 4 day
Minimum Price of  1 on the 1 day
asus
Maximum Price of  4 on the 4 day
Minimum Price of  1 on the 1 day
```

In [ ]:
```

```

In [16]:
```
print("                                                abcd")

file = open("D:\\\para\\abcd.txt","r")
a = file.readline()
print(a)
file.close()
print(" ")
```

```
                                                abcd
Norse or Scandinavian mythology is the body of myths of the North Germanic peop
les, stemming from Norse paganism and continuing after the Christianization of
Scandinavia, and into the Scandinavian folklore of the modern period. The north
ernmost extension of Germanic mythology and stemming from Proto-Germanic folklo
re, Norse mythology consists of tales of various deities, beings, and heroes de
rived from numerous sources from both before and after the pagan period, includ
ing medieval manuscripts, archaeological representations, and folk tradition.
```

In [17]:
```
file =open("D:\\para\\abcd.txt","r")
counter =0
a = file.read()
a.split("\n")
for i in a:
    if i:
        counter +=1
print("No.of lines: ",counter)
```

```
No.of lines:  2988
```

In [19]:
```
num = 0
b= open("D:\\para\\abcd.txt")
for i in b:
    wrd = a.split()
    num += len(wrd)
print("No of words: ",num)
b.close()
```

```
No of words:  3241
```

In [ ]:
```

```

In [23]:
```
print("Find Find frequency of words in a given file:")
fname = input('Enter the file name: ')
try:
    fhand = open(fname)
    counts = dict()
    for line in fhand:
        words = line.split()
        for word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
    print(counts)
except:
    print('File cannot be opened:', fname)
print("\n")
```

```
Find Find frequency of words in a given file:
Enter the file name: D:\\para\\abcd.txt
{'Norse': 6, 'or': 3, 'Scandinavian': 2, 'mythology': 9, 'is': 4, 'the': 47, 'b
ody': 1, 'of': 24, 'myths': 1, 'North': 1, 'Germanic': 3, 'peoples,': 1, 'stemm
ing': 2, 'from': 5, 'paganism': 1, 'and': 26, 'continuing': 1, 'after': 3, 'Chr
istianization': 1, 'Scandinavia,': 1, 'into': 1, 'folklore': 1, 'modern': 3, 'p
eriod.': 1, 'The': 3, 'northernmost': 1, 'extension': 1, 'Proto-Germanic': 1,
'folklore,': 1, 'consists': 2, 'tales': 1, 'various': 1, 'deities,': 1, 'being
s,': 2, 'heroes': 1, 'derived': 1, 'numerous': 3, 'sources': 1, 'both': 2, 'bef
ore': 1, 'pagan': 1, 'period,': 2, 'including': 1, 'medieval': 1, 'manuscript
s,': 1, 'archaeological': 1, 'representations,': 1, 'folk': 1, 'tradition.': 1,
'source': 1, 'texts': 2, 'mention': 1, 'gods,': 1, 'such': 2, 'as': 5, 'hammer-
wielding,': 1, 'humanity-protecting': 1, 'thunder-god': 1, 'Thor,': 1, 'who':
9, 'relentlessly': 1, 'fights': 1, 'his': 1, 'foes;': 1, 'one-eyed,': 1, 'raven
-flanked': 1, 'god': 5, 'Odin,': 1, 'craftily': 1, 'pursues': 1, 'knowledge':
1, 'throughout': 2, 'worlds': 2, 'bestowed': 1, 'among': 2, 'humanity': 2, 'run
ic': 1, 'alphabet;': 1, 'beautiful,': 1, 'seiÃ°r-working,': 1, 'feathered': 1,
'cloak-clad': 1, 'goddess': 4, 'Freyja': 1, 'rides': 1, 'to': 8, 'battle': 2,
'choose': 1, 'slain;': 1, 'vengeful,': 1, 'skiing': 1, 'SkaÃ°i,': 1, 'prefers':
1, 'wolf': 1, 'howls': 1, 'winter': 1, 'mountains': 1, 'seashore;': 1, 'powerfu
l': 1, 'NjÃ¶rÃ°r,': 1, 'may': 3, 'calm': 1, 'sea': 1, 'fire': 1, 'grant': 2, 'w
ealth': 1, 'land;': 1, 'Freyr,': 1, 'whose': 1, 'weather': 1, 'farming': 1, 'as
sociations': 1, 'bring': 1, 'peace': 1, 'pleasure': 1, 'humanity;': 1, 'IÃ°un
n,': 1, 'keeps': 1, 'apples': 1, 'that': 2, 'eternal': 1, 'youthfulness;': 1,
'mysterious': 1, 'Heimdallr,': 1, 'born': 1, 'nine': 1, 'mothers,': 1, 'can':
1, 'hear': 1, 'grass': 1, 'grow,': 1, 'has': 2, 'gold': 1, 'teeth,': 1, 'posses
ses': 1, 'a': 3, 'resounding': 1, 'horn;': 1, "jÃ¶tunn's": 1, 'son,': 1, 'Lok
i,': 1, 'brings': 1, 'tragedy': 1, 'gods': 4, 'by': 1, 'engineering': 1, 'deat
h': 1, "Frigg's": 1, 'beautiful': 1, 'son': 1, 'Baldr;': 1, 'other': 2, 'deitie
s.': 1, 'Most': 1, 'surviving': 2, 'centers': 1, 'on': 1, 'plights': 1, 'thei
r': 2, 'interaction': 1, 'with': 1, 'several': 1, 'jÃ¶tnar,': 1, 'beings': 1,
'be': 5, 'friends,': 1, 'lovers,': 1, 'foes,': 1, 'family': 1, 'members': 1, 'g
ods.': 1, 'cosmos': 1, 'in': 3, 'Nine': 1, 'Worlds': 1, 'flank': 1, 'central':
1, 'sacred': 1, 'tree,': 1, 'Yggdrasil.': 1, 'Units': 1, 'time': 1, 'elements':
2, 'cosmology': 1, 'are': 4, 'personified': 1, 'deities': 1, 'beings.': 1, 'Var
ious': 1, 'forms': 1, 'creation': 1, 'myth': 1, 'recounted,': 1, 'where': 1, 'w
orld': 2, 'created': 1, 'flesh': 1, 'primordial': 1, 'being': 1, 'Ymir,': 1, 'f
irst': 1, 'two': 2, 'humans': 2, 'Ask': 1, 'Embla.': 1, 'These': 1, 'foretold':
1, 'reborn': 2, 'events': 1, 'RagnarÃ¶k': 1, 'when': 2, 'an': 2, 'immense': 1,
'occurs': 1, 'between': 1, 'enemies,': 1, 'enveloped': 1, 'flames,': 1, 'only':
```

```
1, 'anew.': 1, 'There': 1, 'will': 3, 'meet,': 1, 'land': 1, 'fertile': 1, 'gre
en,': 1, 'repopulate': 1, 'world.': 1, 'been': 1, 'subject': 2, 'scholarly': 1,
'discourse': 1, 'since': 1, '17th': 1, 'century,': 1, 'key': 1, 'attracted': 1,
'attention': 1, 'intellectual': 1, 'circles': 1, 'Europe.': 1, 'By': 1, 'way':
1, 'comparative': 1, 'historical': 1, 'linguistics,': 1, 'scholars': 1, 'have':
1, 'identified': 1, 'reaching': 1, 'far': 1, 'back': 1, 'Proto-Indo-European':
1, 'mythology.': 1, 'During': 1, 'Romanticist': 1, 'Viking': 1, 'revival': 1,
're-awoke': 1, 'interest': 1, 'matter,': 1, 'references': 1, 'now': 1, 'found':
1, 'popular': 1, 'culture.': 1}
```

In [25]:
```
print("Show a random line in a file:")
import random
def random_line(fname):
 lines = open(fname).read().splitlines()
 return random.choice(lines)
print(random_line('D:\\para\\abcd.txt'))
```

```
Show a random line in a file:
Norse or Scandinavian mythology is the body of myths of the North Germanic peop
les, stemming from Norse paganism and continuing after the Christianization of
Scandinavia, and into the Scandinavian folklore of the modern period. The north
ernmost extension of Germanic mythology and stemming from Proto-Germanic folklo
re, Norse mythology consists of tales of various deities, beings, and heroes de
rived from numerous sources from both before and after the pagan period, includ
ing medieval manuscripts, archaeological representations, and folk tradition.
```

In [7]:
```
fhand = open('D:\\para\\abcd.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith('From '):
        print(line)
```

In [ ]:
```
```

In [16]:
```
fhand = open("D:\\para\\abcd.txt")
count = 0
for line in fhand:
    line = line.rstrip()
    if line == "": continue
    words = line.split()
    if words[0] !="From": continue
    print(words[1])
    count = count+1
print ("There were", count, "lines in the file with From as the first word")
```

```
There were 0 lines in the file with From as the first word
```

In [ ]:
```
```

In [26]:
```
from csv import writer
def append_list_as_row(file_name, list_of_elem):

    with open('students_mark.csv', 'at', newline='') as write_obj:
        csv_writer = writer(write_obj)
        csv_writer.writerow(list_of_elem)
row_contents = ['dinesh',68,78,89,87,90]
row_contents1 = ['surya',68,78,89,87,90]
row_contents2 = ['hari',68,78,89,87,90]
row_contents3 = ['umesh',68,78,89,87,90]
append_list_as_row('students_mark.csv', row_contents)
append_list_as_row('students_mark.csv', row_contents1)
append_list_as_row('students_mark.csv', row_contents2)
append_list_as_row('students_mark.csv', row_contents3)
```

In [27]:
```
import csv
with open('students_mark.csv', newline ='') as csvfile:
    reader = csv.reader(csvfile, delimiter =' ', quotechar ='1')
    for row in reader:
        print(', '.join(row))
```

```
dinesh,68,78,89,87,90
surya,68,78,89,87,90
hari,68,78,89,87,90
umesh,68,78,89,87,90
```

In [ ]:

In [ ]:

In [ ]: