

INDEX

EXPERIMENT - 1

AIM: TO FIND intersection and union
of two lists

Algorithm

1. Start
2. import numpy
3. declare list a & b
4. enter element into list a & b
5. convert list a,b into array x,y
6. call intersect (x,y)
7. call union (x,y)
8. end

intersect (x,y)

1. declare a list
2. for n in x do
 - 2.1 if n in y then
 - 2.1.1 ext n into list
 - 2.2 end if
3. end for
4. return the list

In []: #Qavindi 333

```
import numpy as np
a=list()
b=list()
n=int(input("Enter Size of A"))
for i in range(0,n):
    a.append(int(input())))
x=np.array(a)
print(x)
m=int(input("Enter Size of B"))
for i in range(0,m):
    b.append(int(input())))
y=np.array(b)
print(y)
print("intersection",np.intersect1d(x,y))
print("union",np.union1d(x,y))
```

Union (x, y)

1. declare list l
2. for n in y do
 - 2.1 if n not in l then
 - 2.1.1 insert n into l
 - 2.2 end if
3. end for
4. return l

Result: programme was successfully executed and output obtained

~~8~~

Output

Enter size of a : 4

8

5

2

0

[8 5 2 0]

Enter size of b : 5

6

2

0

7

8

[6 2 0 7 8]

intersection : [0 2 8]

union: [0 2 5 6 7 8]

Experiment: 2 & 4

Aim: (i) To find count of each word in a text file
 (ii) To find most frequent word in a text file.

1. start

2. open file in S

3. for line in S do

3.1 for t in line.split() do

3.1.1 if t in C then

3.1.1.2 C[t] += 1

3.1.2 else

3.1.2.1 C[t] = 1

3.1.3 end else if

3.2 fixed for

4. end for

5. print C

6. declare maxf = 0

7. for i in C do

7.1 if (maxf < C[i]) then

7.1.1 maxf = C[i]

7.1.2 index = i

7.2 end if

8. end for

In [1]: running ...

```
filename("testfile.txt")
c={}
for lines in file:
    for i in lines.split():
        if i in c:
            c[i]++
        else:
            c[i]=1
print("WORD COUNT DICTIONARY",c)
most=0
for j in c:
    if (most < c[j]):
        most=c[j]
        index=j
print(most," times ",index," has repeated")

WORD COUNT DICTIONARY {'its': 1, 'a': 2, 'nice': 2, 'day.': 2, 'hope': 1, 't': 1, 'have': 1}
2 times a has repeated
```

9. print index,max
10. STOP

Result: output obtained successfully

Experiment - 3

Matrix - Multiplication

Aim: To find product of two matrix

Algorithm

1. start
2. import numpy
3. declare list a & b
4. get no. of rows and columns for matrix $a \times b$ in $m \times m, p \times q$, respectively
5. if $m = p$ then
 - 5.1 print multiplication possible
 - 5.2 Get elements for a
 - 5.3 n = array(a)
 - 5.4 y = a.reshape (n, m)
 - 5.5 print(y)
 - 5.6 Get elements for b
 - 5.7. u = array(b)
 - 5.8 v = u.reshape (p, q)
 - 5.9. print v
 - 5.10 set c = np.zeros((n, q))
 - 5.11 for i in range(n) do
 - 5.11.1 for j in range(q) do
 - 5.11.1.1 $c[i][j] += y[i][k] * v[k][j]$

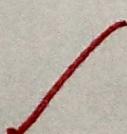
```
In [12]: #Solving 3.3
import numpy as np
a=list()
b=list()
n=int(input("Enter Rows of A"))
m=int(input("Enter Columns of A"))
p=int(input("Enter Rows of B"))
q=int(input("Enter Columns of B"))

if m==p:
    for i in range(n):
        for j in range(m):
            a.append(int(input("Enter Element")))
x=np.array(a)
y=x.reshape(n,m)
print(y)
for i in range(p):
    for j in range(q):
        b.append(int(input("Enter Element")))
u=np.array(b)
v=u.reshape(p,q)
print(v)

#MULTIPLY
print(np.multiply(y,v))

else:
    print("Invalid Input")
```

Enter Rows of A1
Enter Columns of A2
Enter Rows of B2
Enter Columns of B1
Enter Element1
Enter Element2
[[1 2]]
Enter Element3
Enter Element4
[[3]
 [4]]
[[3 6]
 [4 8]]



5.11.1-2 end for

5.11.2 end for

5.12 end for

5.13 print C

else

6.1 print (multiplication not possible)

Result: programme was successfully executed and output obtained