ai-program-day-4

May 5, 2024

[1]:
```
l=[10,20,30,10,50,20,'yu',6,'yu']
d=[]
for i in l:
    if l.count(i)>1 and i not in d:
        d.append(i)
print(d)
```

```
[10, 20, 'yu']
```

[2]:
```
d=[]
for i in range(len(l)):
    for j in range(i+1,len(l)):
        if l[i]==l[j]:
            d.append(l[i])
print(d)
```

```
[10, 20, 'yu']
```

[3]:
```
for j in range(5):
    for i in range(5):
        print('*',end=' ')
```

```
* * * * * * * * * * * * * * * * * * * * * * * * *
```

[4]:
```
for j in range(5):
    for i in range(5):
        print('*',end='  ')
    print()
```

```
*  *  *  *  *
*  *  *  *  *
*  *  *  *  *
*  *  *  *  *
*  *  *  *  *
```

[5]:
```
for j in range(5):
    for i in range(j+1):
        print('*',end=' ')
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

[6]:

```
for j in range(5):
    for i in range(j,5):
        print('*',end=' ')
    print()
```

```
* * * * *
* * * *
* * *
* *
*
```

[7]:

```
n=5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')
    for j in range(i+1):
        print('0',end=' ')
    print()
```

```
          0
        0 0
      0 0 0
    0 0 0 0
  0 0 0 0 0
```

[8]:

```
for i in range(5,1,-1):
    for j in range(5,1,-1):
        print(i,end=' ')
    print()
```

```
5 5 5 5
4 4 4 4
3 3 3 3
2 2 2 2
```

[9]:

```
n=5
for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print('*',end=' ')
    print()
```

```
  * * * * *
    * * * *
      * * *
        * *
          *
```

[10]:

```
n=5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')
    for j in range(i+1):
        print('*',end=' ')
    for j in range(i):
        print('*',end=' ')
    print()
```

```
          *
        * * *
      * * * * *
    * * * * * * *
  * * * * * * * * *
```

[11]:

```
n=5
for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print('*',end=' ')
    for j in range(i,n-1):
        print('*',end=' ')
    print()
```

```
  * * * * * * * * *
    * * * * * * *
      * * * * *
        * * *
          *
```

[12]:

```
n=5
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')
    for j in range(i+1):
        print('*',end=' ')
    for j in range(i):
        print('*',end=' ')
    print()


for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print('*',end=' ')
    for j in range(i,n-1):
        print('*',end=' ')
    print()
```

```
         *
       * * *
     * * * * *
   * * * * * * *
 * * * * * * * * *
   * * * * * * *
     * * * * *
       * * *
         *
```

[42]:

```
data={'Jay':['male',2000],'Kiran':['male',1999],'Vijay':
 ↪['male',1998],'Sanskriti':['female',2002]}
count=0
for i in data:
    if isinstance(data[i],list):
        count+=1
print(count)
```

```
4
```

[47]:

```
count=0
for i in range(len(data)):
    if type(data[i])==str:
        count+=1
        i+=1
print(count)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[47], line 3
      1 count=0
      2 for i in range(len(data)):
----> 3     if type(data[i])==str:
      4         count+=1
      6 print(count)

KeyError: 0
```

[44]:
```
data['Jay']
```

[44]:
```
['male', 2000]
```

[45]:
```
data.values()
```

[45]:
```
dict_values([['male', 2000], ['male', 1999], ['male', 1998], ['female', 2002]])
```

[46]:
```
data.iloc[2]
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[46], line 1
----> 1 data.iloc[2]

AttributeError: 'dict' object has no attribute 'iloc'
```

[43]:
```
dataitems=list(data.items())
```

[44]:
```
dataitems
```

[44]:
```
[('Jay', ['male', 2000]),
 ('Kiran', 'male'),
 ('Vijay', ['male', 1998]),
 ('Sanskriti', ['female', 2002])]
```

[47]:
```
type(dataitems[0][1])
```

[47]:
```
list
```

[48]:
```
len(dataitems[1])
```

[48]:
```
2
```

[50]:
```
count=0
for i in range (len(dataitems)):
    if type(dataitems[i][1])==list:
        count+=1
print(count)
```

```
3
```

[57]:

```
def factorial(n):
    if n==1:
        return 1
    else:
        return(n*factorial(n-1))
print(factorial(5))
```

```
120
```

[60]:

```
num=int(input("number for factorial?"))
factorial=1
for i in range(1,num+1):
    factorial=factorial*i
print(num,'!',"=",factorial)
```

```
number for factorial? 7

7 ! = 5040
```

[63]:

```
prime=[]
for i in range(2,20):
    c=0
    for j in range(1,i):
        if i%j==0:
            c+=1
    if c==1:
        prime.append(i)
print(prime)
print(sum(prime))
```

```
[2, 3, 5, 7, 11, 13, 17, 19]
77
```

[64]:

```
n=int(input("Number?"))
for i in range(2,n):
    if n%i==0:
        print('not a prime number')
        break
    else:
        print('Prime Number')
        braek
```

```
Number? 4

not a prime number
```

[82]:

```
s=input()
if s==s[::-1]:
    print('Palindrome')
```

```
else:
    print('not')
```

```
 hghj
```

```
not
```

[102]:

```
str1=input()
list1=str1.split(' ')
print(list1)
list2=list1[::-1]
print(list2)
str2=' '.join(list2)
print(str2)
```

```
 Bairavi Rajashree
```

```
['Bairavi', 'Rajashree']
['Rajashree', 'Bairavi']
Rajashree Bairavi
```

[114]:

```
mylist=[1,2,2,2,3,3,4,5,5,6,6,7,8]
new_list=[]
for i in mylist:
    if mylist.count(i)==1:
        new_list.append(i)
print(new_list)
```

```
[1, 4, 7, 8]
```

[115]:

```
[i for i in mylist if mylist.count(i)==1]
```

[115]:

```
[1, 4, 7, 8]
```

[50]:

```
str1="a,a,a,b,b,c,c,c"
list1=str1.split(',')
count_ele=[]
str_ele=[]
for ch in list1:
    if not ch in str_ele:
        count_ele.append(f"{ch}:{list1.count(ch)}")
        str_ele.append(ch)
print(",".join(count_ele))
```

```
a:3,b:2,c:3
```

[53]:

```
b=[[6,8],[90,4],[23,67]]
c=[2,8,6,9,10,23,14]
b.sort(key=lambda x:x[0])
```

```
print(b)
print(sorted(c))
```

```
[[6, 8], [23, 67], [90, 4]]
[2, 6, 8, 9, 10, 14, 23]
```

[56]:

```
try:
    n=int(input("How many numbers?"))
except:
    print("Please enter number")
else:
    list_num=[]
    try:
        for i in range(n):
            num=int(input())
            list_num.append(num)
        mean=lambda x: sum(x)/len(x)
        print(mean(list_num))
    except:
        print("Some error occurred")
```

```
How many numbers? 4
 5
 6
 7
 8

6.5
```

[57]:

```
2+3+4+6+6+76
```

[57]:

```
97
```

[58]:

```
20/5
```

[58]:

```
4.0
```

[59]:

```
n=5
p=1
for i in range(n):
    for j in range(i+1):
        print(p,end=' ')
    p+=1
    print()
```

```
1
2 2
3 3 3
```

```
4 4 4 4
5 5 5 5 5
```

[63]:

```
n=5
p=n
for i in range(n):
    for j in range(i+1):
        print(p,end=' ')
    p-=1
    print()
```

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

[64]:

```
n=5
p=0
for i in range(n):
    for j in range(i+1):
        print(p,end=' ')
    p+=2
    print()
```

```
0
2 2
4 4 4
6 6 6 6
8 8 8 8 8
```

[65]:

```
n=5
for i in range(n):
    for j in range(i+1):
        if (i%2==0):
            print(1,end=' ')
        else:
            print(2,end=' ')
    print()
```

```
1
2 2
1 1 1
2 2 2 2
1 1 1 1 1
```

[67]:

```
n=5
p=1
```

```
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')
    for j in range(i+1):
        print(p,end=' ')
    for j in range(i):
        print(p,end=' ')
    p+=1
    print()


for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print(p,end=' ')
    for j in range(i,n-1):
        print(p,end=' ')
    p+=1
    print()
```

```
          1
        2 2 2
      3 3 3 3 3
    4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5
    6 6 6 6 6 6 6
      7 7 7 7 7
        8 8 8
          9
```

[68]:

```
n=5
p=1
for i in range(n-1):
    for j in range(i,n):
        print(' ',end=' ')
    for j in range(i+1):
        print(p,end=' ')
    for j in range(i):
        print(p,end=' ')
    p+=1
    print()


for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print(p,end=' ')
    for j in range(i,n-1):
        print(p,end=' ')
    p-=1
    print()
```

```
          1
        2 2 2
      3 3 3 3 3
    4 4 4 4 4 4 4
  5 5 5 5 5 5 5 5 5
    4 4 4 4 4 4 4
      3 3 3 3 3
        2 2 2
          1
```

[69]:

```
n=5
for i in range(n):
    p=1
    for j in range(i+1):
        print(p,end=' ')
        p+=1
    print()
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

[71]:

```
n=5
for i in range(n):
    for j in range(i+1):
        print(' ',end=' ')
    p=1
    for j in range(i,n):
        print(p,end=' ')
        p+=1
    print()
```

```
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

[72]:

```
n=5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')
    p=1
    for j in range(i+1):
        print(p,end=' ')
        p+=1
    for j in range(i):
        print(p,end=' ')
    print()
```

```
          1
        1 2 3
      1 2 3 4 4
    1 2 3 4 5 5 5
  1 2 3 4 5 6 6 6 6
```

[73]:

```
n=5
for i in range(n):
    p=n
    for j in range(i+1):
        print(p,end=' ')
        p-=1
    print()
```

```
5
5 4
5 4 3
5 4 3 2
5 4 3 2 1
```

[75]:

```
n=5
for i in range(n):
    p=n
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print(p,end=' ')
        p-=1
    print()
```

```
  5 4 3 2 1
    5 4 3 2
      5 4 3
        5 4
          5
```

[76]:

```
n=5
k=n
for i in range(n):
    p=k
    for j in range(i+1):
        print(' ',end=' ')
    for j in range(i,n):
        print(p,end=' ')
        p-=1
    k-=1
    print()
```

```
  5 4 3 2 1
    4 3 2 1
      3 2 1
        2 1
          1
```

[77]:

```
n=5
for i in range(n):
    for j in range(i,n):
        print(' ',end=' ')
    p=1
    for j in range(i):
        print(p,end=' ')
        p+=1
    for j in range(i+1):
        print(p,end=' ')
        p-=1
    print()
```

```
          1
        1 2 1
      1 2 3 2 1
    1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
```

[78]:

```
n=5
p=1
for i in range(n):
    for j in range(i+1):
        print(p,end=' ')
        p+=1
    print()
```

```
1
2 3
4 5 6
```

```
7 8 9 10
11 12 13 14 15
```

[102]:

```
#create the function that will captilize the first letter of first name and␣
 ↪last name
#input: bairavi rajashree
#expected output:Bairavi Rajashree
x='bairavi'
y='rajashree'
print(x.capitalize(),y.capitalize())
```

```
Bairavi Rajashree
```

[103]:

```
def cap_names(x):
    a=x.title()
    return a
cap_names("bairavi rajashree")
```

[103]:

```
'Bairavi Rajashree'
```

[104]:

```
def cap_names(x):
    a=x.split(' ')
    a=(i.capitalize() for i in a)
    return ' '.join(a)
cap_names("bairavi rajashree")
```

[104]:

```
'Bairavi Rajashree'
```

[105]:

```
def cap_names(x):
    a=x.split(' ')
    b=[]
    for i in range(len(a)):
        b.append(a[i].capitalize())
    return ' '.join(b)
print(cap_names("bairavi rajashree"))
```

```
Bairavi Rajashree
```

[106]:

```
#create a data frame with 4-5 names and apply this function on that dat frame␣
 ↪to give proper case names.
import pandas as pd
data={"Names":["bairavi rajashree","gopu tejashwini","barigela␣
 ↪vijayalaxmi","ballari bhavana","bommakanti meghana"]}
df=pd.DataFrame(data)
```

[107]:

```
type(df["Names"])
```

[107]: pandas.core.series.Series

[108]:
```
df["Names"][1]
```

[108]: 'gopu tejashwini'

[109]:
```
print(cap_names(df["Names"][1]))
```

```
Gopu Tejashwini
```

[110]:
```
for i in range(len(df["Names"])):
    print(cap_names(df["Names"][i]))
```

```
Bairavi Rajashree
Gopu Tejashwini
Barigela Vijayalaxmi
Ballari Bhavana
Bommakanti Meghana
```

[111]:
```
list1=[]
for i in range(len(df["Names"])):
    list1.append(cap_names(df["Names"][i]))
print(list1)
```

```
['Bairavi Rajashree', 'Gopu Tejashwini', 'Barigela Vijayalaxmi', 'Ballari
Bhavana', 'Bommakanti Meghana']
```

[112]:
```
df['Names']=list1
```

[113]:
```
df
```

[113]:
```
                  Names
0     Bairavi Rajashree
1       Gopu Tejashwini
2  Barigela Vijayalaxmi
3       Ballari Bhavana
4    Bommakanti Meghana
```

[114]:
```
import calendar
```

[115]:
```
class Employee:
    comp="DELL"
    depart="WIAI"
    def wish(self):
        print("Good Morining")
    def add(self,a,b):
        return(a+b)
```

[116]:
```
X=Employee()
```

[117]:
```
X.comp
```

[117]:
```
'DELL'
```

[118]:
```
X.wish()
```

```
Good Morining
```

[119]:
```
X.add(5,7)
```

[119]:
```
12
```

[120]:
```
20+X.add(1,4)
```

[120]:
```
25
```

[121]:
```
X.depart
```

[121]:
```
'WIAI'
```

[122]:
```
class Employee:
    comp="LLF"
    depart="WIAI"
    def __init__(self,name):
        print(f"{name},Welcome to LLF")
    def wish(self):
        print("Good Morining")
    def add(self,a,b):
        return(a+b)
```

[123]:
```
Employee("Sonu")
```

```
Sonu,Welcome to LLF
```

[123]:
```
<__main__.Employee at 0x22d40c9fb90>
```

[124]:
```
class Employee:
    comp="LLF"
    depart="WIAI"
    def wish(self):
        print("Good Morining")
    def add(self,a,b):
        return(a+b)
class Employee2(Employee):
    comp="google"
    location="delhi"
```

[125]:

```
y=Employee()
```

[126]:

```
y.comp
```

[126]:

```
'LLF'
```

[127]:

```
y.wish()
```

```
Good Morining
```

[128]:

```
y.add(4,4)
```

[128]:

```
8
```

[129]:

```
class Employee:
    comp="LLF"
    depart="WIAI"
    def wish(self):
        print("Good Morining")
    def add(self,a,b):
        return(a+b)
class Employee2(Employee):
    comp="google"
    location="delhi"
    def exp(self,a,b):
        return(a**b)
```

[130]:

```
x=Employee()
```

[131]:

```
y=Employee2()
```

[132]:

```
x.add(4,4)
```

[132]:

```
8
```

[133]:

```
y.exp(4,4)
```

[133]:

```
256
```

[158]:

```
#create a class with name classroom,
#add functions to add new students or remove students
#add function to collect email id/contact details
class classroom:
    def __init__(self):
        self.students=[]
        self.contacts=[]
    def add_student(self,name):
        s=name
        self.students.append(s)
    def add_contact(self,email_id):
        e=email_id
        self.contacts.append(e)
    def remove_stud(self,name):
        self.students.remove(name)
```

[159]:
```
x=classroom()
```

[160]:
```
x.add_student("Rajashree")
```

[161]:
```
x.add_student("Yutika")
```

[162]:
```
x.students
```

[162]:
```
['Rajashree', 'Yutika']
```

[163]:
```
x.remove_stud("Yutika")
```

[164]:
```
x.add_student("Tejashwini")
```

[165]:
```
x.students
```

[165]:
```
['Rajashree', 'Tejashwini']
```

[166]:
```
#create a shopping cart app which will take the items &give total bill to␣
 ↪customer.
```

[167]:
```
pip install rake_nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: rake_nltk in
c:\users\janam\appdata\roaming\python\python311\site-packages (1.0.6)
Requirement already satisfied: nltk<4.0.0,>=3.6.2 in
c:\programdata\anaconda3\lib\site-packages (from rake_nltk) (3.8.1)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-
packages (from nltk<4.0.0,>=3.6.2->rake_nltk) (8.1.7)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-
packages (from nltk<4.0.0,>=3.6.2->rake_nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in
c:\programdata\anaconda3\lib\site-packages (from nltk<4.0.0,>=3.6.2->rake_nltk)
(2023.10.3)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-
packages (from nltk<4.0.0,>=3.6.2->rake_nltk) (4.65.0)
```

```
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-
packages (from click->nltk<4.0.0,>=3.6.2->rake_nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

[168]:

```
pip install yake
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: yake in
c:\users\janam\appdata\roaming\python\python311\site-packages (0.4.8)
Requirement already satisfied: tabulate in c:\programdata\anaconda3\lib\site-
packages (from yake) (0.9.0)
Requirement already satisfied: click>=6.0 in c:\programdata\anaconda3\lib\site-
packages (from yake) (8.1.7)
Requirement already satisfied: numpy in
c:\users\janam\appdata\roaming\python\python311\site-packages (from yake)
(1.26.4)
Requirement already satisfied: segtok in
c:\users\janam\appdata\roaming\python\python311\site-packages (from yake)
(1.5.11)
Requirement already satisfied: networkx in c:\programdata\anaconda3\lib\site-
packages (from yake) (3.1)
Requirement already satisfied: jellyfish in c:\programdata\anaconda3\lib\site-
packages (from yake) (1.0.1)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-
packages (from click>=6.0->yake) (0.4.6)
Requirement already satisfied: regex in c:\programdata\anaconda3\lib\site-
packages (from segtok->yake) (2023.10.3)
Note: you may need to restart the kernel to use updated packages.
```

[169]:

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\janam\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[169]:

```
True
```

[170]:

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\janam\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[170]:

```
True
```

[171]:

```
from rake_nltk import Rake
rake_nltk_var=Rake()
text="""I am a programmer from India, and I am here to guide you
with Data Science, Machine Learning, Python, and C++ for free.
I hope you will learning a lot in your journey towards coding,
Machine Learning and Artificial Intelligence with me"""
rake_nltk_var.extract_keywords_from_text(text)
keyword_extracted=rake_nltk_var.get_ranked_phrases()
print(keyword_extracted)
```

```
['journey towards coding', 'data science', 'c ++', 'artificial intelligence',
'machine learning', 'machine learning', 'learning', 'python', 'programmer',
'lot', 'india', 'hope', 'guide', 'free']
```

[172]:

```
import yake
kw_extractor=yake.KeywordExtractor()
text="""I am a programmer from India, and I am here to guide you
with Data Science, Machine Learning, Python, and C++ for free.
I hope you will learning a lot in your journey towards coding,
Machine Learning and Artificial Intelligence with me"""
language="en"
max_ngram_size=3
deduplication_threshold=0.9
numOfKeywords=20
custom_kw_extractor=yake.KeywordExtractor(lan=language, n=max_ngram_size,␣
 ↪dedupLim=deduplication_threshold, top=numOfKeywords, features=None)
keywords=custom_kw_extractor.extract_keywords(text)
for kw in keywords:
    print(kw)
```

```
('Data Science', 0.00643837423010947)
('programmer from India', 0.015764304412048798)
('Machine Learning', 0.029576920898664524)
('Python', 0.04698714696179021)
('India', 0.06943226792709996)
('Science', 0.06943226792709996)
('Machine', 0.0912104349482713)
('Data', 0.09213564440274201)
('Learning', 0.10205355376741022)
('Artificial Intelligence', 0.13009370396381506)
('free', 0.1555727842192338)
('programmer', 0.2235221233674397)
('guide', 0.2235221233674397)
('journey towards coding', 0.24235067029814455)
('Learning and Artificial', 0.29318751048925845)
('Artificial', 0.339289884392853)
('Intelligence', 0.339289884392853)
```

```
('coding', 0.3901915207694476)
('learning a lot', 0.4143808302423282)
('hope', 0.49994496517870207)
```