

This screenshot shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer lists various Python files, including '1Dictionary.py'. The code editor displays the following Python code:

```
1 # retrieve the marks of specific student #SANJANA 24CS212
2 a={'Ali': 90, 'Sara': 85, 'Tariq': 78, 'Aisha': 92}
3 b=input("Enter the name of the student:")
4 print(a[b])
5
6 # Count the occurrence of each word using string #SANJANA 24CS212
7 a='apple banana apple orange banana apple'
8 b=a.split()
9 c={}
10 for i in b:
11     if i in c:
12         c[i]+=1
13     else:
14         c[i]=1
15 print(c)
16
17 # merge two dictionaries if they have same key sum #SANJANA 24CS212
18 a={'a':1,'b':2,'c':3}
19 b={'a':4,'b':5}
20 c={}
21 for i in a:
22     if i in b:
23         c[i]=a[i]+b[i]
24     else:
25         c[i]=a[i]
26 for i in b:
27     if i not in c:
28         c[i]=b[i]
29 print(c)
30
31
```

This screenshot shows a VS Code editor window with a file explorer on the left and a code editor in the center. The file explorer lists various Python files, including '1Dictionary.py'. The code editor displays the following Python code:

```
32 # find the character that appears the most in a string print the maximum occurs #SANJANA 24CS212
33 a="banana"
34 b={}
35 for i in a:
36     if i in b:
37         b[i]+=1
38     else:
39         b[i]=1
40 print(max(b,key=b.get))
41 print(max(b.values()))
42
43 # find the key that are present in both dictionaries #SANJANA 24CS212
44 a={'a':1,'b':2,'c':3}
45 b={'a':4,'b':5}
46 c={}
47 for i in a:
48     if i in b:
49         c[i]=a[i]
50 print(c)
51
52 #Given a list of words, return the k most frequent words, sorted by frequency and
53 # lexicographical order. #SANJANA 24CS212
54 a=['i','love','leetcode','i','love','coding']
55 b=2
56 c={}
57 for i in a:
58     if i in c:
59         c[i]+=1
60     else:
61         c[i]=1
62
```

This screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer lists 24 Python files under the 'PYTHON PROGRAM' folder. The code editor displays the content of '1Dictionary.py', which contains two Python scripts. The first script, starting at line 65, is for finding anagrams from a list of words. The second script, starting at line 76, is a 'Real-Time Weather Data Analysis System' that processes weather data for New York, London, and Tokyo, finding the best, hottest, and coldest cities based on various metrics like AQI and temperature.

```
65 # 7. Given a list of words, group anagrams together. #SANJANA 24CS212
66 a=["eat","tea","tan","ate","nat","bat"]
67 b={}
68 for i in a:
69     c=''.join(sorted(i))
70     if c in b:
71         b[c].append(i)
72     else:
73         b[c]=[i]
74 print(list(b.values()))
75
76 # Real-Time Weather Data Analysis System #SANJANA 24CS212
77 weather_data = {
78     "New York": {"Temperature": 25, "Humidity": 60, "AQI": 80},
79     "London": {"Temperature": 18, "Humidity": 70, "AQI": 75},
80     "Tokyo": {"Temperature": 30, "Humidity": 55, "AQI": 90}
81 }
82 best_city = min(weather_data, key=lambda city: weather_data[city]["AQI"])
83 print("Best Air Quality:", best_city)
84 max_temp_city = max(weather_data, key=lambda city: weather_data[city]["Temperature"])
85 min_temp_city = min(weather_data, key=lambda city: weather_data[city]["Temperature"])
86 print("Hottest City:", max_temp_city)
87 print("Coldest City:", min_temp_city)
88 for city, info in weather_data.items():
89     if info["AQI"] > 100:
90         print(f"{city}: Poor Air Quality Alert!")
91     elif info["Temperature"] > 35:
92         print(f"{city}: Heatwave Alert!")
93     elif info["Humidity"] > 80:
94         print(f"{city}: High Humidity Alert!")
95
```

This screenshot shows the same VS Code editor window with the '1Dictionary.py' file open. The code editor now displays a third Python script, starting at line 99, titled '# IoT-Based Smart Home Energy Optimization System'. This script defines a dictionary 'e' containing energy usage data for various appliances (LR, AC, TV, BR, F, K, O). It then processes this data to find the maximum energy usage, print it, and generate a list of recommendations ('rec') for each appliance. The script also includes logic to provide tips or auto-off suggestions based on the usage data.

```
99 # IoT-Based Smart Home Energy Optimization System# IoT Smart Home Energy Optimize - SANJANA 24CS212
100 e = {
101     "LR": {"L": 50, "AC": 500, "TV": 100}, "BR": {"L": 30, "AC": 600, "F": 50}, "K": {"Fz": 300, "O": 200, "L": 40}
102 }
103 m = {}
104 for r, a in e.items():
105     for n, u in a.items():
106         if n in m:
107             m[n] += u
108         else:
109             m[n] = u
110 mx = max(m, key=m.get)
111 print("Max Energy App:", mx)
112 print("Usage:", m[mx])
113 rec = {
114     "L": "Use LED", "AC": "Set 24°C", "TV": "Turn Off Unused", "F": "Use Efficient Fan", "Fz": "Keep Optimal Temp",
115     "O": "Use Microwave"
116 }
117 for n in m:
118     if n in rec:
119         print(f"Tip for {n}: {rec[n]}")
120     else:
121         print(f"No Tip for {n}")
122 u = {
123     "LR": {"TV": 100}, "BR": {"F": 50}, "K": {"O": 200}
124 }
125 for r, a in u.items():
126     for n, v in a.items():
127         if v < 100:
128             print(f"Auto-OFF {n} in {r}")
129
```

```
File Edit Selection View Go Run Terminal Help Python_Program
EXPLORER
PYTHON PROGRAM
1_Separator.py
1Dictionary.py
2_Even_Odd.py
2a_Vowel_Consonant.py
2b_SumEven.py
2c_Factorial.py
2listSorting.py
2nd_Largest.py
3_C_to_F.py
4_Area_of_Triangle.py
5_Circum_area_circle.py
6_Quadratic.py
7_Square_root.py
8_Multiple_5.py
9_Multiple_5_7.py
10_swap.py
11_add2Num.py
12_floatnum.py
13_Factorial.py
14_Sum_n.py
15_Random.py
16_Calendar.py
17_Positive_negative.py
18_LargestOf_3.py
19_MultiplicationTable.py
20_Palindrom.py
21_Reverse.py
22_LeapYear.py
23_SquareNum.py
24_Vowel_Consonant.py
OUTLINE
TIMELINE
master* 0.0.0 Live Share

1Dictionary.py
132 #Automated Resume Shortlisting for Job Applications #SANJANA 24CS212
133 r = {
134     "C1": {"S": ["Py", "AI", "ML"], "E": 5},
135     "C2": {"S": ["Jv", "C1", "D0"], "E": 3},
136     "C3": {"S": ["Py", "DS", "SQL"], "E": 4}
137 }
138 print("Exp 5+ :")
139 for n, d in r.items():
140     if d["E"] >= 5:
141         print(n)
142 print("\nDS Role :")
143 for n, d in r.items():
144     if "DS" in d["S"]:
145         print(n)
146 req = ["Py", "DS", "SQL"]
147 print("\nSkill Rank :")
148 rk = {}
149 for n, d in r.items():
150     m = len(set(d["S"]) & set(req))
151     p = (m / len(req)) * 100
152     rk[n] = p
153 for n, s in sorted(rk.items(), key=lambda x: x[1], reverse=True):
154     print(f"{n} - {s}%")
155
156 # 11. Multi-Store Inventory and Auto-Restock System #SANJANA 24CS212
157 s = {
158     "S1": {"L": {"stk": 5, "pr": 700}, "P": {"stk": 10, "pr": 500}},
159     "S2": {"L": {"stk": 2, "pr": 750}, "P": {"stk": 5, "pr": 550}}
160 }
161 p = "L"
162 mp = 9999
```

```
File Edit Selection View Go Run Terminal Help Python_Program
EXPLORER
PYTHON PROGRAM
1_Separator.py
1Dictionary.py
2_Even_Odd.py
2a_Vowel_Consonant.py
2b_SumEven.py
2c_Factorial.py
2listSorting.py
2nd_Largest.py
3_C_to_F.py
4_Area_of_Triangle.py
5_Circum_area_circle.py
6_Quadratic.py
7_Square_root.py
8_Multiple_5.py
9_Multiple_5_7.py
10_swap.py
11_add2Num.py
12_floatnum.py
13_Factorial.py
14_Sum_n.py
15_Random.py
16_Calendar.py
17_Positive_negative.py
18_LargestOf_3.py
19_MultiplicationTable.py
20_Palindrom.py
21_Reverse.py
22_LeapYear.py
23_SquareNum.py
24_Vowel_Consonant.py
OUTLINE
TIMELINE
master* 0.0.0 Live Share

1Dictionary.py
165 if p in v and v[p]["pr"] < mp:
166     mp = v[p]["pr"]
167     sn = k
168 print("Low Price:", sn, mp)
169 for k, v in s.items():
170     for i, d in v.items():
171         if d["stk"] < 3:
172             print("Restock:", i, "at", k)
173
174 # 12. Decentralized Cryptocurrency Wallet System #SANJANA 24CS212
175 w = {
176     "A": {"B": 0.5, "E": 2, "D": 5000},
177     "B": {"B": 1, "E": 5, "D": 10000},
178     "C": {"B": 0.2, "E": 1, "D": 3000}
179 }
180 r = {"B": 30000, "E": 2000, "D": 0.1}
181 w["A"]["B"] -= 0.1
182 w["B"]["B"] += 0.1
183 t = {}
184 for u, c in w.items():
185     t[u] = sum(q * r[cc] for cc, q in c.items())
186 top = sorted(t.items(), key=lambda x: x[1], reverse=True)[:2]
187 print("Top:", top)
188
189 # 13. Fake News Detection Using Word Frequency Analysis #SANJANA 24CS212
190 word_frequencies = {
191     "article1": {"COVID": 10, "Vaccine": 8, "Cure": 12},
192     "article2": {"Election": 15, "Fraud": 20, "Democracy": 5},
193     "article3": {"Investment": 18, "Scam": 22, "Profit": 7}
194 }
```

```
195 }
196 suspicious = []
197 for article, words in word_frequencies.items():
198     if "Fraud" in words or "Scam" in words:
199         suspicious.append(article)
200 print("Suspicious Articles:", suspicious)
201
202 # 14. Airline Ticket Pricing and Availability System #SANJANA 24CS212
203 f = {
204     "A1": {"E": {"s": 50, "p": 500}, "B": {"s": 10, "p": 1500}},
205     "B2": {"E": {"s": 20, "p": 700}, "B": {"s": 5, "p": 2000}}
206 }
207 f["A1"]["E"]["s"] -= 1
208 print("Booked")
209 for k, v in f.items():
210     for c, d in v.items():
211         if d["s"] < 10:
212             d["p"] *= 1.2
213 print(f)
214
215 # 15. Advanced Student Performance Analytics #SANJANA 24CS212
216 s = {
217     "A": {"M": 85, "P": 78, "C": 92},
218     "B": {"M": 90, "P": 88, "C": 79},
219     "C": {"M": 76, "P": 85, "C": 85}
220 }
221
222 for n, m in s.items():
223     a = sum(m.values()) / len(m)
224     print(n, "Avg:", a)
225     sub = ["M", "P", "C"]
226     for sb in sub:
227         t = max(s, key=lambda x: s[x][sb])
228         print("Top in", sb, ":", t)
229     for n, m in s.items():
230         a = sum(m.values()) / len(m)
231         if a < 80:
232             print(n, "Finnous")
```