

Assignment 4: Implementing Control Structures

Nitish Dhinakaran

005036295

Github:- https://github.com/ndhinakaran36295/MSCS-632_Assignment4_Control-Structures

Schedule Output:-

I ran the Python and JavaScript code from the above repo, using the given the input data in

https://github.com/ndhinakaran36295/MSCS-632_Assignment4_Control-Structures/blob/c409124d4fe82161e35df154aa265d26c1568d0f/employee_scheduler.js#L123-L214

```
107 // Add employees with their shift preferences and priority rankings
108 employee_data = {
109     "Employee A": {
110         "Monday": ["morning", "afternoon", "evening"],
111         "Tuesday": ["morning", "afternoon", "evening"],
112         "Wednesday": ["morning", "afternoon", "evening"],
113         "Thursday": ["morning", "afternoon", "evening"],
114         "Friday": ["morning", "afternoon", "evening"],
115         "Saturday": ["morning", "afternoon", "evening"],
116         "Sunday": ["morning", "afternoon", "evening"]
117     },
118     "Employee B": {
119         "Monday": ["afternoon", "morning", "evening"],
120         "Tuesday": ["afternoon", "morning", "evening"],
121         "Wednesday": ["afternoon", "morning", "evening"],
122         "Thursday": ["afternoon", "morning", "evening"],
123         "Friday": ["afternoon", "morning", "evening"],
124         "Saturday": ["afternoon", "morning", "evening"],
125         "Sunday": ["afternoon", "morning", "evening"]
126     },
127     "Employee C": {
128         "Monday": ["evening", "morning", "afternoon"],
129         "Tuesday": ["evening", "morning", "afternoon"],
130         "Wednesday": ["evening", "morning", "afternoon"],
131         "Thursday": ["evening", "morning", "afternoon"],
132         "Friday": ["evening", "morning", "afternoon"],
133         "Saturday": ["evening", "morning", "afternoon"],
134         "Sunday": ["evening", "morning", "afternoon"]
135     },
136     "Employee D": {
137         "Monday": ["morning", "evening", "afternoon"],
138         "Tuesday": ["morning", "evening", "afternoon"],
139         "Wednesday": ["morning", "evening", "afternoon"],
140         "Thursday": ["morning", "evening", "afternoon"],
141         "Friday": ["morning", "evening", "afternoon"],
142         "Saturday": ["morning", "evening", "afternoon"],
143         "Sunday": ["morning", "evening", "afternoon"]
144     },
145     "Employee E": {
146         "Monday": ["afternoon", "evening", "morning"],
147         "Tuesday": ["afternoon", "evening", "morning"],
148         "Wednesday": ["afternoon", "evening", "morning"],
149         "Thursday": ["afternoon", "evening", "morning"],
150         "Friday": ["afternoon", "evening", "morning"],
151         "Saturday": ["afternoon", "evening", "morning"],
152         "Sunday": ["afternoon", "evening", "morning"]
153     },
154 }
```

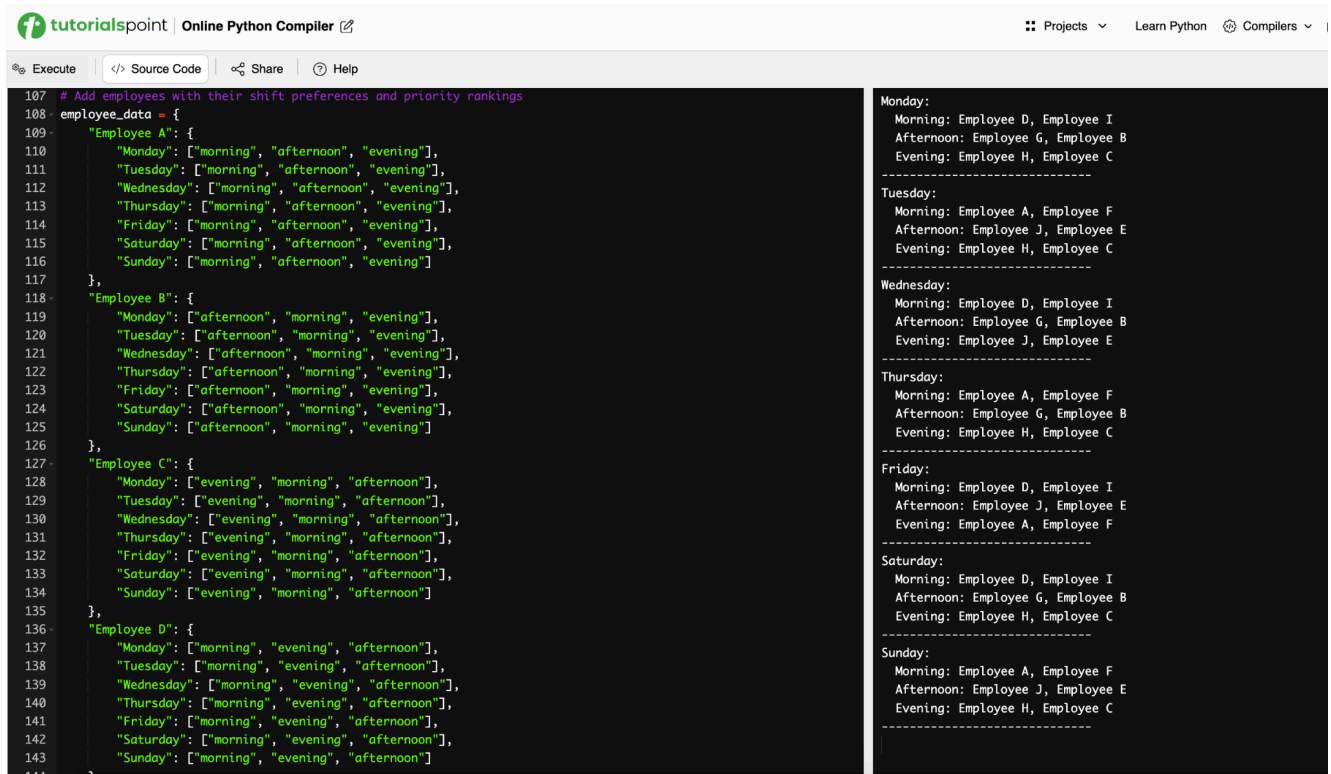
```

153 },
154 "Employee F": {
155     "Monday": ["morning", "afternoon", "evening"],
156     "Tuesday": ["morning", "afternoon", "evening"],
157     "Wednesday": ["morning", "afternoon", "evening"],
158     "Thursday": ["morning", "afternoon", "evening"],
159     "Friday": ["morning", "afternoon", "evening"],
160     "Saturday": ["morning", "afternoon", "evening"],
161     "Sunday": ["morning", "afternoon", "evening"]
162 },
163 "Employee G": {
164     "Monday": ["afternoon", "morning", "evening"],
165     "Tuesday": ["afternoon", "morning", "evening"],
166     "Wednesday": ["afternoon", "morning", "evening"],
167     "Thursday": ["afternoon", "morning", "evening"],
168     "Friday": ["afternoon", "morning", "evening"],
169     "Saturday": ["afternoon", "morning", "evening"],
170     "Sunday": ["afternoon", "morning", "evening"]
171 },
172 "Employee H": {
173     "Monday": ["evening", "morning", "afternoon"],
174     "Tuesday": ["evening", "morning", "afternoon"],
175     "Wednesday": ["evening", "morning", "afternoon"],
176     "Thursday": ["evening", "morning", "afternoon"],
177     "Friday": ["evening", "morning", "afternoon"],
178     "Saturday": ["evening", "morning", "afternoon"],
179     "Sunday": ["evening", "morning", "afternoon"]
180 },
181 "Employee I": {
182     "Monday": ["morning", "evening", "afternoon"],
183     "Tuesday": ["morning", "evening", "afternoon"],
184     "Wednesday": ["morning", "evening", "afternoon"],
185     "Thursday": ["morning", "evening", "afternoon"],
186     "Friday": ["morning", "evening", "afternoon"],
187     "Saturday": ["morning", "evening", "afternoon"],
188     "Sunday": ["morning", "evening", "afternoon"]
189 },
190 "Employee J": {
191     "Monday": ["afternoon", "evening", "morning"],
192     "Tuesday": ["afternoon", "evening", "morning"],
193     "Wednesday": ["afternoon", "evening", "morning"],
194     "Thursday": ["afternoon", "evening", "morning"],
195     "Friday": ["afternoon", "evening", "morning"],
196     "Saturday": ["afternoon", "evening", "morning"],
197     "Sunday": ["afternoon", "evening", "morning"]
198 }
199 }

```

As shown in the above input, I chose 10 employees to work in the company and the scheduler handles all the required conditions and generates a schedule for the employees. The output is shown below

Output:-



The screenshot shows the 'tutorialspoint Online Python Compiler' interface. The left pane contains Python code for a scheduling algorithm, and the right pane shows the resulting schedule for seven days of the week.

```
107 # Add employees with their shift preferences and priority rankings
108 employee_data = {
109     "Employee A": {
110         "Monday": ["morning", "afternoon", "evening"],
111         "Tuesday": ["morning", "afternoon", "evening"],
112         "Wednesday": ["morning", "afternoon", "evening"],
113         "Thursday": ["morning", "afternoon", "evening"],
114         "Friday": ["morning", "afternoon", "evening"],
115         "Saturday": ["morning", "afternoon", "evening"],
116         "Sunday": ["morning", "afternoon", "evening"]
117     },
118     "Employee B": {
119         "Monday": ["afternoon", "morning", "evening"],
120         "Tuesday": ["afternoon", "morning", "evening"],
121         "Wednesday": ["afternoon", "morning", "evening"],
122         "Thursday": ["afternoon", "morning", "evening"],
123         "Friday": ["afternoon", "morning", "evening"],
124         "Saturday": ["afternoon", "morning", "evening"],
125         "Sunday": ["afternoon", "morning", "evening"]
126     },
127     "Employee C": {
128         "Monday": ["evening", "morning", "afternoon"],
129         "Tuesday": ["evening", "morning", "afternoon"],
130         "Wednesday": ["evening", "morning", "afternoon"],
131         "Thursday": ["evening", "morning", "afternoon"],
132         "Friday": ["evening", "morning", "afternoon"],
133         "Saturday": ["evening", "morning", "afternoon"],
134         "Sunday": ["evening", "morning", "afternoon"]
135     },
136     "Employee D": {
137         "Monday": ["morning", "evening", "afternoon"],
138         "Tuesday": ["morning", "evening", "afternoon"],
139         "Wednesday": ["morning", "evening", "afternoon"],
140         "Thursday": ["morning", "evening", "afternoon"],
141         "Friday": ["morning", "evening", "afternoon"],
142         "Saturday": ["morning", "evening", "afternoon"],
143         "Sunday": ["morning", "evening", "afternoon"]
144     }
145 }
```

The output on the right shows the schedule for each day:

Monday:
Morning: Employee D, Employee I
Afternoon: Employee G, Employee B
Evening: Employee H, Employee C

Tuesday:
Morning: Employee A, Employee F
Afternoon: Employee J, Employee E
Evening: Employee H, Employee C

Wednesday:
Morning: Employee D, Employee I
Afternoon: Employee G, Employee B
Evening: Employee J, Employee E

Thursday:
Morning: Employee A, Employee F
Afternoon: Employee G, Employee B
Evening: Employee H, Employee C

Friday:
Morning: Employee D, Employee I
Afternoon: Employee J, Employee E
Evening: Employee A, Employee F

Saturday:
Morning: Employee D, Employee I
Afternoon: Employee G, Employee B
Evening: Employee H, Employee C

Sunday:
Morning: Employee A, Employee F
Afternoon: Employee J, Employee E
Evening: Employee H, Employee C

As you can see above, the scheduler comes up with a schedule for all the employees trying to maximize satisfying the preferences and the required conditions for the schedule (all shifts and days are covered, no employee works multiple shifts a day, no employee works more than 5 days a week). This program also ensures every employee gets more or less an equal number of shifts.