

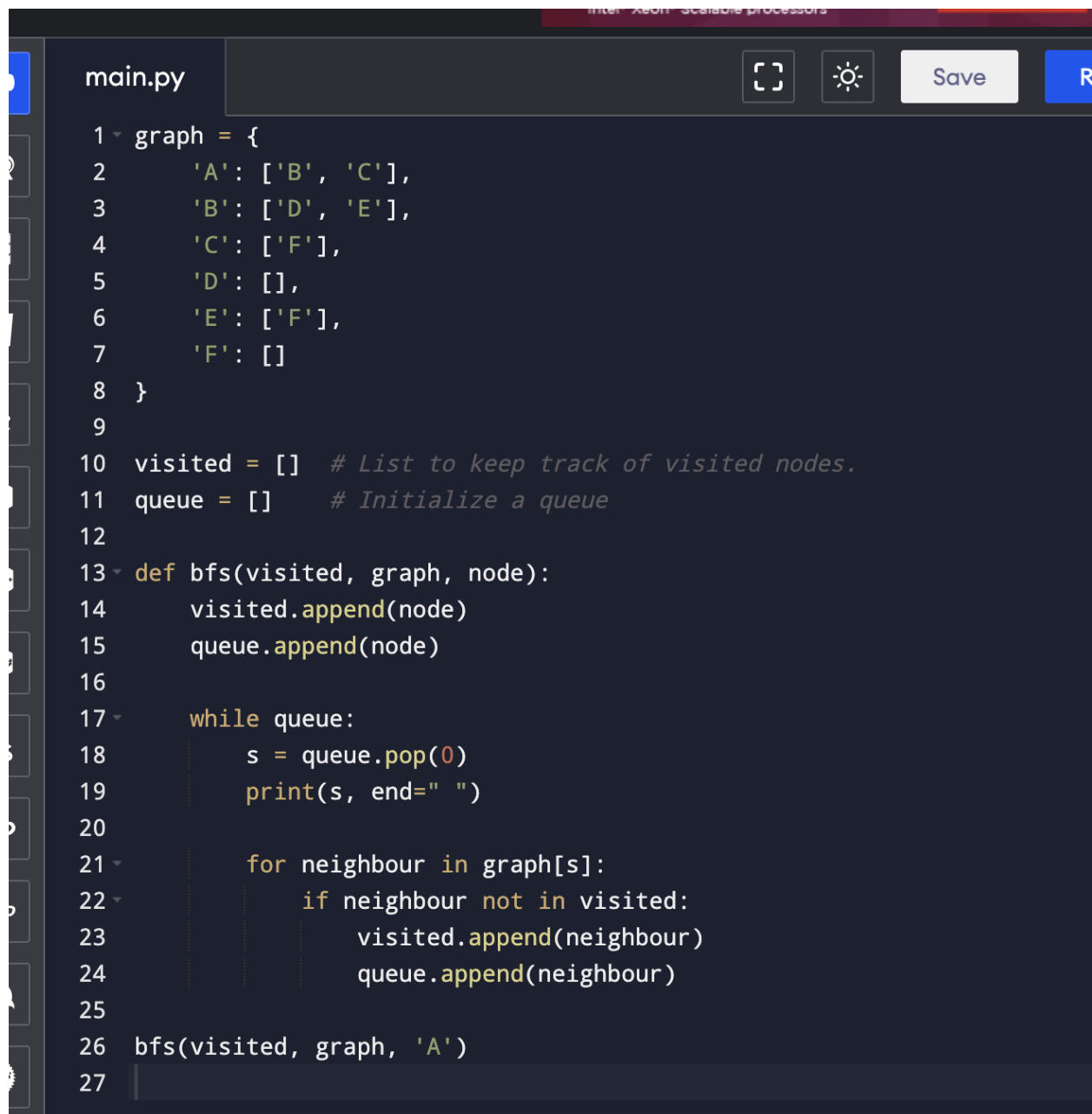
Assignment 5:

Name : Darain shahedi

Admission NO: I21ma010

Q1) Implement Breath first search in python.

Source code:

A screenshot of a code editor window titled 'main.py'. The editor has a dark theme and a sidebar on the left. The code implements a Breadth First Search (BFS) algorithm. It starts with a graph definition, initializes a visited list and a queue, defines a bfs function, and then calls bfs with the graph and starting node 'A'.

```
1 graph = {
2     'A': ['B', 'C'],
3     'B': ['D', 'E'],
4     'C': ['F'],
5     'D': [],
6     'E': ['F'],
7     'F': []
8 }
9
10 visited = [] # List to keep track of visited nodes.
11 queue = []   # Initialize a queue
12
13 def bfs(visited, graph, node):
14     visited.append(node)
15     queue.append(node)
16
17     while queue:
18         s = queue.pop(0)
19         print(s, end=" ")
20
21         for neighbour in graph[s]:
22             if neighbour not in visited:
23                 visited.append(neighbour)
24                 queue.append(neighbour)
25
26 bfs(visited, graph, 'A')
27
```

Output

```
Shell
A B C D E F >
```

Q2) Implementation of depth first search in python:

```
main.py
1 graph = {
2     'A': ['B', 'C'],
3     'B': ['D', 'E'],
4     'C': ['F'],
5     'D': [],
6     'E': ['F'],
7     'F': []
8 }
9
10 visited = set() # Set to keep track of visited nodes.
11
12 def dfs(visited, graph, node):
13     if node not in visited:
14         print(node)
15         visited.add(node)
16         for neighbour in graph[node]:
17             dfs(visited, graph, neighbour)
18
19 dfs(visited, graph, 'A')
20
```

Output:

Run	Shell	Clear
	A B D E F C >	