Continuous Assessment – 1

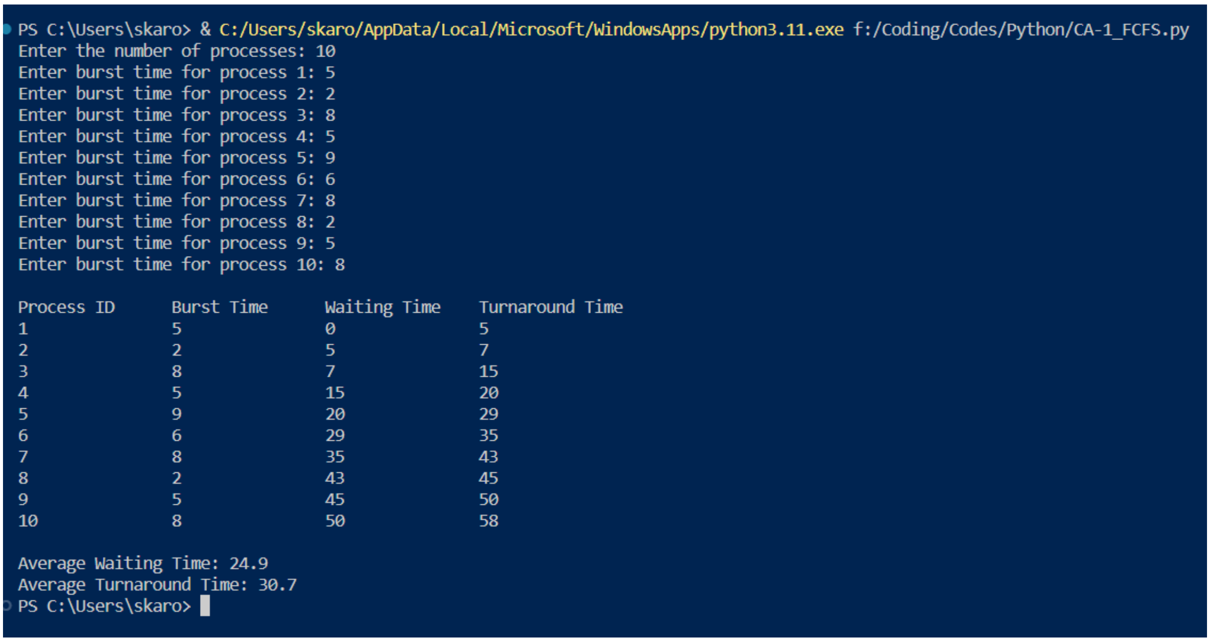
Aromal S Kunnel – 22BAI10288

FCFS

CODE :

```
def fcfs(n):

  processes = []
  for i in range(n):
    process = {"pid": i + 1, "bt": int(input(f"Enter burst time for process {i
+ 1}: "))}
    processes.append(process)


  processes[0]["wt"] = 0
  processes[0]["tt"] = processes[0]["bt"] + processes[0]["wt"]
  for i in range(1, n):
    processes[i]["wt"] = processes[i - 1]["tt"]
    processes[i]["tt"] = processes[i]["bt"] + processes[i]["wt"]

  total_wt = sum(process["wt"] for process in processes)
  total_tt = sum(process["tt"] for process in processes)

  avg_wt = total_wt / n
  avg_tt = total_tt / n

  print("\nProcess ID\tBurst Time\tWaiting Time\tTurnaround Time")
  for process in processes:
    print(f"{process['pid']}\t\t{process['bt']}\t\t{process['wt']}\t\t{process
['tt']}")
  print(f"\nAverage Waiting Time: {avg_wt}")
  print(f"Average Turnaround Time: {avg_tt}")

n = int(input("Enter the number of processes: "))

fcfs(n)
```

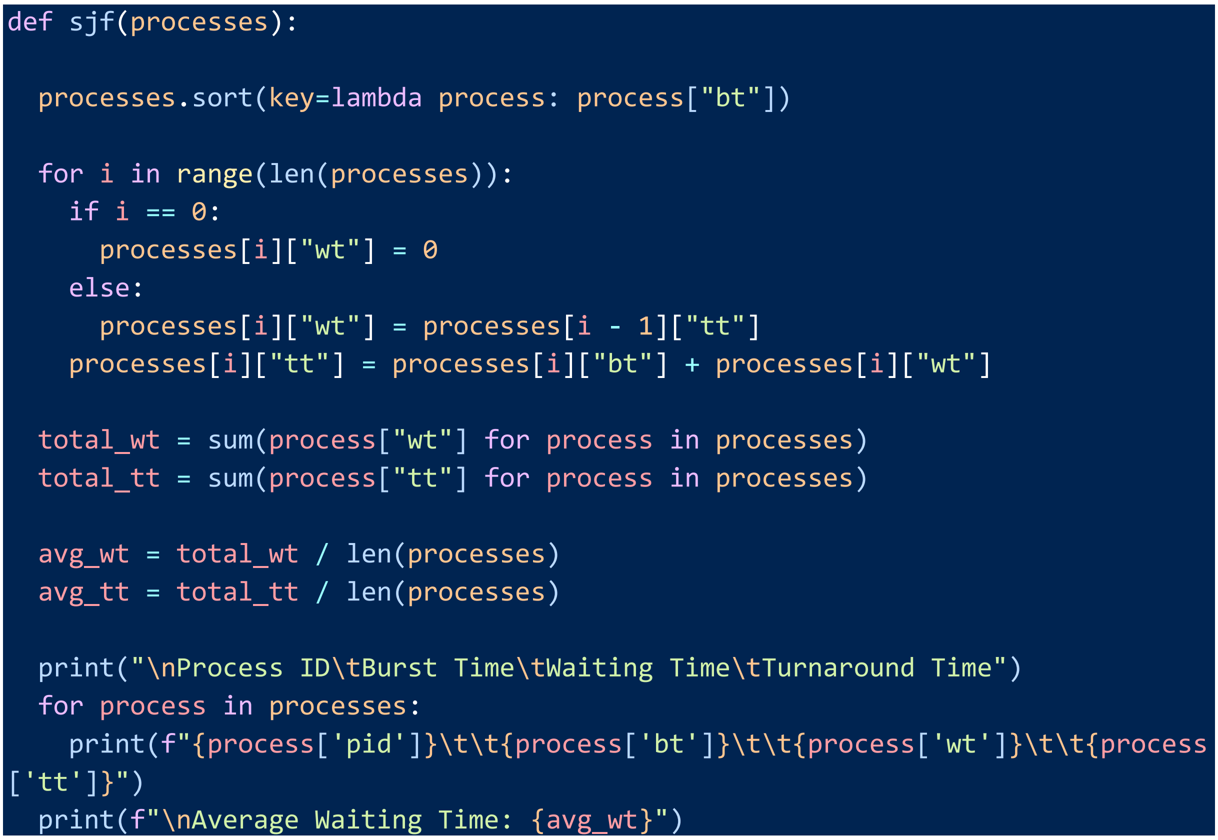
OUTPUT :



```
PS C:\Users\skaro> & C:/Users/skaro/AppData/Local/Microsoft/WindowsApps/python3.11.exe f:/Coding/Codes/Python/CA-1_FCFS.py
Enter the number of processes: 10
Enter burst time for process 1: 5
Enter burst time for process 2: 2
Enter burst time for process 3: 8
Enter burst time for process 4: 5
Enter burst time for process 5: 9
Enter burst time for process 6: 6
Enter burst time for process 7: 8
Enter burst time for process 8: 2
Enter burst time for process 9: 5
Enter burst time for process 10: 8

Process ID      Burst Time      Waiting Time    Turnaround Time
1               5               0               5
2               2               5               7
3               8               7               15
4               5               15              20
5               9               20              29
6               6               29              35
7               8               35              43
8               2               43              45
9               5               45              50
10              8               50              58

Average Waiting Time: 24.9
Average Turnaround Time: 30.7
PS C:\Users\skaro> █
```

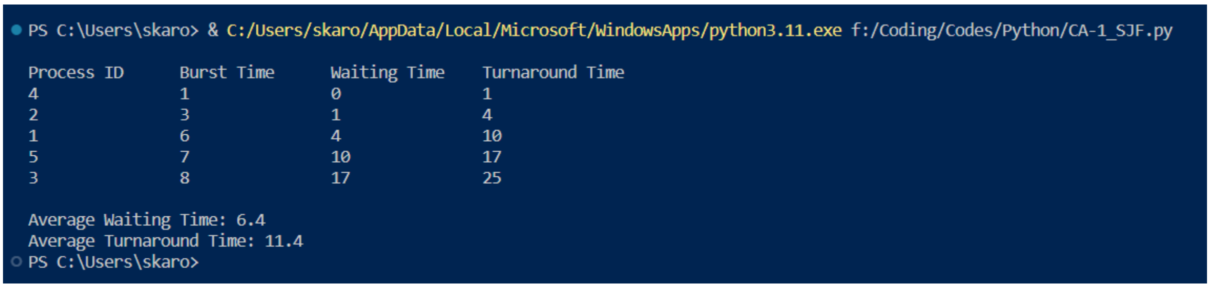
SJF

CODE :



```
def sjf(processes):

  processes.sort(key=lambda process: process["bt"])

  for i in range(len(processes)):
    if i == 0:
      processes[i]["wt"] = 0
    else:
      processes[i]["wt"] = processes[i - 1]["tt"]
    processes[i]["tt"] = processes[i]["bt"] + processes[i]["wt"]

  total_wt = sum(process["wt"] for process in processes)
  total_tt = sum(process["tt"] for process in processes)

  avg_wt = total_wt / len(processes)
  avg_tt = total_tt / len(processes)

  print("\nProcess ID\tBurst Time\tWaiting Time\tTurnaround Time")
  for process in processes:
    print(f"{process['pid']}\t\t{process['bt']}\t\t{process['wt']}\t\t{process
['tt']}")
  print(f"\nAverage Waiting Time: {avg_wt}")
  print(f"Average Turnaround Time: {avg_tt}")

processes = [
    {"pid": 1, "bt": 6},
    {"pid": 2, "bt": 3},
    {"pid": 3, "bt": 8},
    {"pid": 4, "bt": 1},
    {"pid": 5, "bt": 7},
]

sjf(processes)
```

OUTPUT :



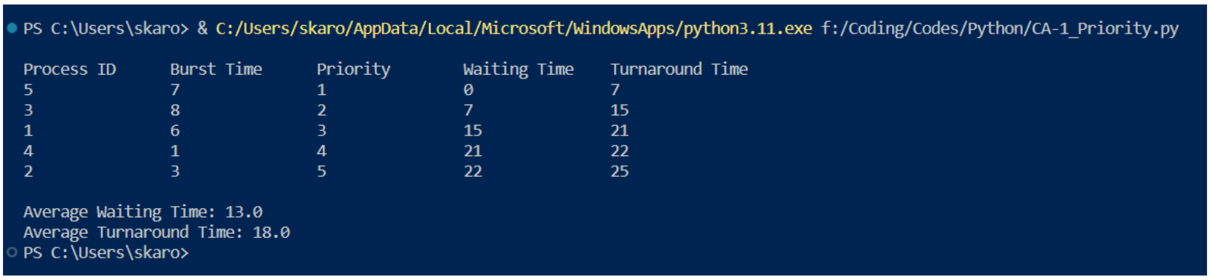
```
● PS C:\Users\skaro> & C:/Users/skaro/AppData/Local/Microsoft/WindowsApps/python3.11.exe f:/Coding/Codes/Python/CA-1_SJF.py

  Process ID      Burst Time      Waiting Time    Turnaround Time
  4               1               0               1
  2               3               1               4
  1               6               4               10
  5               7               10              17
  3               8               17              25

  Average Waiting Time: 6.4
  Average Turnaround Time: 11.4
○ PS C:\Users\skaro>
```

PRIORITY SCHEDULING

CODE :

```
def priority(processes):

  processes.sort(key=lambda process: process["prior"])

  for i in range(len(processes)):
    if i == 0:
      processes[i]["wt"] = 0
    else:
      processes[i]["wt"] = processes[i - 1]["tt"]
    processes[i]["tt"] = processes[i]["bt"] + processes[i]["wt"]

  total_wt = sum(process["wt"] for process in processes)
  total_tt = sum(process["tt"] for process in processes)

  avg_wt = total_wt / len(processes)
  avg_tt = total_tt / len(processes)
  print("\nProcess ID\tBurst Time\tPriority\tWaiting Time\tTurnaround Time")
  for process in processes:
    print(f"{process['pid']}\t\t{process['bt']}\t\t{process['prior']}\t\t{process['wt']}\t\t{process['tt']}")
  print(f"\nAverage Waiting Time: {avg_wt}")
  print(f"Average Turnaround Time: {avg_tt}")

processes = [
    {"pid": 1, "bt": 6, "prior": 3},
    {"pid": 2, "bt": 3, "prior": 5},
    {"pid": 3, "bt": 8, "prior": 2},
    {"pid": 4, "bt": 1, "prior": 4},
    {"pid": 5, "bt": 7, "prior": 1},
]

priority(processes)
```

OUTPUT :



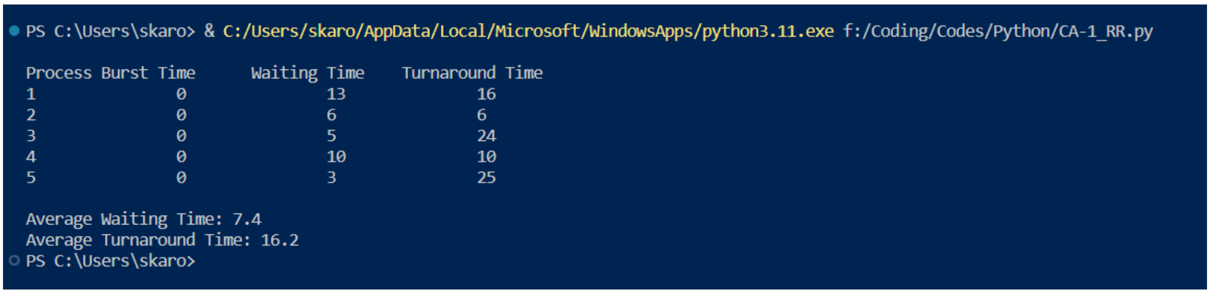
```
● PS C:\Users\skaro> & C:/Users/skaro/AppData/Local/Microsoft/WindowsApps/python3.11.exe f:/Coding/Codes/Python/CA-1_Priority.py

  Process ID      Burst Time      Priority        Waiting Time    Turnaround Time
  5               7               1               0               7
  3               8               2               7               15
  1               6               3               15              21
  4               1               4               21              22
  2               3               5               22              25

  Average Waiting Time: 13.0
  Average Turnaround Time: 18.0
○ PS C:\Users\skaro>
```

RR

CODE :

```
def round_robin(processes, time_quantum):

  n = len(processes)
  completed = [False] * n
  waiting_time = [0] * n
  turnaround_time = [0] * n
  current_time = 0

  max_burst_time = max(process["bt"] for process in processes)

  while not all(completed):
    for i in range(n):
      if not completed[i] and processes[i]["bt"] > 0:
        executed = min(processes[i]["bt"], time_quantum)

        # Update process data
        processes[i]["bt"] -= executed
        current_time += executed

        if processes[i]["bt"] == 0:
          completed[i] = True
          turnaround_time[i] = current_time
          waiting_time[i] = turnaround_time[i] - processes[i]["rt"]

      if processes[i]["rt"] <= current_time and not completed[i]:
        processes[i]["rt"] = current_time

  average_waiting_time = sum(waiting_time) / n
  average_turnaround_time = sum(turnaround_time) / n

  print("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time")
  for i in range(n):
    print(f"{i +
1}\t\t{processes[i]['bt']}\t\t{waiting_time[i]}\t\t{turnaround_time[i]}")
  print(f"\nAverage Waiting Time: {average_waiting_time}")
  print(f"Average Turnaround Time: {average_turnaround_time}")

processes = [
    {"bt": 6, "rt": 2},
    {"bt": 3, "rt": 0},
    {"bt": 8, "rt": 1},
    {"bt": 1, "rt": 0},
    {"bt": 7, "rt": 2},
]

time_quantum = 3

round_robin(processes, time_quantum)
```

OUTPUT :



```
PS C:\Users\skaro> & C:/Users/skaro/AppData/Local/Microsoft/WindowsApps/python3.11.exe f:/Coding/Codes/Python/CA-1_RR.py

Process Burst Time      Waiting Time    Turnaround Time
1               0               13              16
2               0               6               6
3               0               5               24
4               0               10              10
5               0               3               25

Average Waiting Time: 7.4
Average Turnaround Time: 16.2
PS C:\Users\skaro>
```