CPU Scheduling

Team Members:-
(Student’s name): Lateefa Ali Alsfri
ID No.: 2110642

(Student’s name): Majd Ahmed Khawjah
ID No.: 2110117

Task Breakup

Code	All Together
Presentation 1 - 4	Majd
Presentation 5 - 8	Lateefa
Over All Edit	Majd
Format	Lateefa

Code

```
for (i = 0, c = 'A'; i < n; i++, c++)
  {
     Q1[i].name = c;
     printf("\nEnter the arrival time of process %c: ", Q1[i].name);
     scanf("%d", &Q1[i].AT);

     printf("\nEnter the burst time of process %c: ", Q1[i].name);
     scanf("%d",  &Q1[i].BT);

     Q1[i].RT = Q1[i].BT;
     ART += Q1[i].BT;

  }

  sortByArrival();
  t = Q1[0].AT;
  printf("First queue Process following RR with qt=8");
  printf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");
  for (i = 0; i < n; i++)
  {
```

```
for (i = 0; i < n; i++)
{

    if (Q1[i].RT <= qua1)
    {

        t += Q1[i].RT;
        Q1[i].RT = 0;
        Q1[i].WT = t - Q1[i].AT - Q1[i].BT;
        Q1[i].TAT = t - Q1[i].AT;
        printf("\n%c\t\t%d\t\t%d\t\t%d", Q1[i].name, Q1[i].BT, Q1[i].WT, Q1[i].TAT);
        AWT += Q1[i].WT;
        ATAT += Q1[i].TAT;
    }

    if (Q1[i].RT > qua1)
    {
        Q2[k].WT = t;
        t += qua1;
        Q1[i].RT -= qua1;

        Q1[i].WT = t - Q1[i].AT - qua1;
        Q1[i].TAT = t - Q1[i].AT;
        printf("\n%c\t\t%d\t\t%d\t\t%d", Q1[i].name, qua1, Q1[i].WT, Q1[i].TAT);

        Q2[k].BT = Q1[i].RT;
        Q2[k].RT = Q2[k].BT;
        Q2[k].name = Q1[i].name;
        k = k + 1;
        flag = 16;
    }
}
```

```
if (flag == 16) {
      printf("\nSecond queue Process following RR with qt=16");
      printf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");

      for (i = 0; i < k; i++){
         if (Q2[i].RT <= qua2){
            t += Q2[i].RT;
            Q2[i].RT = 0;
            Q2[i].WT = t - Q1[i].AT - qua1 - Q2[i].BT;
            Q2[i].TAT = t - Q1[i].AT;
            printf("\n%c\t\t%d\t\t%d\t\t%d", Q2[i].name, Q2[i].BT,
Q2[i].WT, Q2[i].TAT);
            AWT += Q2[i].WT;
            ATAT += Q2[i].TAT;
         }

         if (Q2[i].RT > qua2)  {
            Q3[r].AT = t;
            t += qua2;
            Q2[i].RT -= qua2;

            Q2[i].WT = t - Q1[i].AT - qua1 - qua2;
            Q2[i].TAT = t - Q1[i].AT;
            printf("\n%c\t\t%d\t\t%d\t\t%d", Q2[i].name, qua2,
Q2[i].WT, Q2[i].TAT);

            Q3[r].BT = Q2[i].RT;
            Q3[r].RT = Q3[r].BT;
            Q3[r].name = Q2[i].name;
            r = r + 1;
            flag = 1;
         }
      }
   }
```

```
 if (flag == 1) {
      printf("\nThird queue Process following FCFS ");
      printf("\nProcess\t\tRT\t\tWT\t\tTAT\t\t");

      for (i = 0; i < r; i++) {
         int qua3 = Q3[i].BT;
         t += qua3;

         Q3[i].TAT = t - Q1[i].AT;
         Q3[i].WT = t - Q1[i].AT - qua1 - qua2 - qua3;
         printf("\n%c\t\t%d\t\t%d\t\t%d\t\t", Q3[i].name,
Q3[i].BT, Q3[i].WT, Q3[i].TAT);
         AWT += Q3[i].WT;
         ATAT += Q3[i].TAT;
      }
   }

   printf("\n\nAverage WT: %f\n", (float)AWT / (float)n);
   printf("Average TAT: %f\n", (float)ATAT / (float)n);
   printf("Average RT: %f\n", (float)ART / (float)n);
}
```

The Output

```
Enter the number of processes to do:3

Enter the arrival time of process A: 0

Enter the burst time of process A: 29

Enter the arrival time of process B: 3

Enter the burst time of process B: 29

Enter the arrival time of process C: 1

Enter the burst time of process C: 29
First queue Process following RR with qt=8
Process         RT              WT              TAT
A               8               0               8
C               8               7               15
B               8               13              21
Second queue Process following RR with qt=16
Process         RT              WT              TAT
A               16              16              40
C               16              31              55
B               16              45              69
Third queue Process following FCFS
Process         RT              WT              TAT
A               5               48              77
C               5               52              81
B               5               55              84

Average WT: 51.666668
Average TAT: 80.666664
Average RT: 29.000000


Response Time A: 0
Response Time C: 7
Response Time B: 13


...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter the number of processes to do:-1
Please enter number of processes more than zero to scheduling:0
Please enter number of processes more than zero to scheduling:1

Enter the arrival time of process A: 2

Enter the burst time of process A: 10
First queue Process following RR with qt=8
Process         RT              WT              TAT
A               8               0               8
Second queue Process following RR with qt=16
Process         RT              WT              TAT
A               2               0               10

Average WT: 0.000000
Average TAT: 10.000000
Average RT: 10.000000


Response Time A: 0


...Program finished with exit code 0
Press ENTER to exit console.
```

Features and Capabilities

1

Our project take the number of desired processes as an input from the user.

2

Show the output in a menu like format.

3

The program is an example of MFQ compute the processes in the order of:
1st- RR quantum 8.
2nd- RR quantum 16.
3rd- FCFS.

4

The program has a sorting method that sort processes by AT (Arrival Time).

User Manual

1	• User asked to enter the number of desired processes (should be more than zero).
2	• User asked to enter the arrival time of the process (for each process).
3	• User asked to enter the burst time of the process (for each process).
4	• User then get the menu view of computing results.

THANK YOU!

DO YOU HAVE ANY QUESTIONS?