**4. Implement a simulation of the round robin scheduling algorithm where the program should accept the number of processes, their arrival times, burst times, the time quantum from the user and perform the execution of these processes according to the round robin algorithm**

#include<stdio.h>

int main()

{

int i, limit, total = 0, x, counter = 0, time\_quantum;

int wait\_time = 0, turnaround\_time = 0, arrival\_time[10], burst\_time[10], temp[10];

float average\_wait\_time, average\_turnaround\_time;

printf("\nEnter Total Number of Processes:");

scanf("%d", &limit);

x = limit;

for(i = 0; i < limit; i++)

{

printf("\nEnter Details of Process[%d]\n", i + 1);

printf("Arrival Time:");

scanf("%d", &arrival\_time[i]);

printf("Burst Time:");

scanf("%d", &burst\_time[i]);

temp[i] = burst\_time[i];

}

printf("\nEnter Time Quantum:\t");

scanf("%d", &time\_quantum);

printf("\nProcess ID\t\tBurst Time\t Turnaround Time\t Waiting Time\n");

for(total = 0, i = 0; x != 0;)

{

if(temp[i] <= time\_quantum && temp[i] > 0)

{

total = total + temp[i];

temp[i] = 0;

counter = 1;

}

if(temp[i] > 0)

{

temp[i] = temp[i] - time\_quantum;

total = total + time\_quantum;

}

if(temp[i] == 0 && counter == 1)

{

x--;

printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i + 1, burst\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]);

wait\_time = wait\_time + total - arrival\_time[i] - burst\_time[i];

turnaround\_time = turnaround\_time + total - arrival\_time[i];

counter = 0;

}

if(i == limit - 1)

{

i = 0;

}

else if(arrival\_time[i + 1] <= total)

{

i++;

}

else

{

i = 0;

}

}

average\_wait\_time = wait\_time \* 1.0 / limit;

average\_turnaround\_time = turnaround\_time \* 1.0 / limit;

printf("\nAverage Waiting Time:\t%2f", average\_wait\_time);

printf("\nAvg Turnaround Time:\t%2f\n", average\_turnaround\_time);

return 0;

}

INPUT:

1. Enter Total Number of Processes:4

Enter Details of Process[1]

Arrival Time:0

Burst Time:4

Enter Details of Process[2]

Arrival Time:1

Burst Time:7

Enter Details of Process[3]

Arrival Time:2

Burst Time:5

Enter Details of Process[4]

Arrival Time:3

Burst Time:6

Enter Time Quantum: 3

Process ID Burst Time Turnaround Time Waiting Time

Process[1] 4 13 9

Process[3] 5 16 11

Process[4] 6 18 12

Process[2] 7 21 14

Average Waiting Time: 11.500000

Avg Turnaround Time: 17.000000

INPUT:

Enter Total Number of Processes:5

Enter Details of Process[1]

Arrival Time:0

Burst Time:8

Enter Details of Process[2]

Arrival Time:0

Burst Time:2

Enter Details of Process[3]

Arrival Time:0

Burst Time:7

Enter Details of Process[4]

Arrival Time:0

Burst Time:3

Enter Details of Process[5]

Arrival Time:0

Burst Time:5

Enter Time Quantum: 3

Process ID Burst Time Turnaround Time Waiting Time

Process[2] 2 5 3

Process[4] 3 11 8

Process[5] 5 22 17

Process[1] 8 24 16

Process[3] 7 25 18

Average Waiting Time: 12.400000

Avg Turnaround Time: 17.400000

{

temp[i] = temp[i] - time\_quantum;

total = total + time\_quantum;

}

if(temp[i] == 0 && counter == 1)

{

x--;

printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i + 1, burst\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]);

wait\_time = wait\_time + total - arrival\_time[i] - burst\_time[i];

turnaround\_time = turnaround\_time + total - arrival\_time[i];

counter = 0;

}

if(i == limit - 1)

{

i = 0;

}

else if(arrival\_time[i + 1] <= total)

{

i++;

}

else

{

i = 0;

}

}

average\_wait\_time = wait\_time \* 1.0 / limit;

average\_turnaround\_time = turnaround\_time \* 1.0 / limit;

printf("\nAverage Waiting Time:\t%2f", average\_wait\_time);

printf("\nAvg Turnaround Time:\t%2f\n", average\_turnaround\_time);

return 0;

}

INPUT:

1. Enter Total Number of Processes:4

Enter Details of Process[1]

Arrival Time:0

Burst Time:4

Enter Details of Process[2]

Arrival Time:1

Burst Time:7

Enter Details of Process[3]

Arrival Time:2

Burst Time:5

Enter Details of Process[4]

Arrival Time:3

Burst Time:6

Enter Time Quantum: 3

Process ID Burst Time Turnaround Time Waiting Time

Process[1] 4 13 9

Process[3] 5 16 11

Process[4] 6 18 12

Process[2] 7 21 14

Average Waiting Time: 11.500000

Avg Turnaround Time: 17.000000

INPUT:

Enter Total Number of Processes:5

Enter Details of Process[1]

Arrival Time:0

Burst Time:8

Enter Details of Process[2]

Arrival Time:0

Burst Time:2

Enter Details of Process[3]

Arrival Time:0

Burst Time:7

Enter Details of Process[4]

Arrival Time:0

Burst Time:3

Enter Details of Process[5]

Arrival Time:0

Burst Time:5

Enter Time Quantum: 3

Process ID Burst Time Turnaround Time Waiting Time

Process[2] 2 5 3

Process[4] 3 11 8

Process[5] 5 22 17

Process[1] 8 24 16

Process[3] 7 25 18

Average Waiting Time: 12.400000

Avg Turnaround Time: 17.400000