17. Write a program to compute the average waiting time and average turnaround time based on Round Robin scheduling for the following process with the given CPU burst times and quantum time slots 4 ms, ( and the assumption that all jobs arrive at the same time.)

Process Burst Time

P1 24

P2 3

P3 3

Program:

#include<stdio.h> #include<stdlib.h>

void round\_robin(int bt[], int n, int quantum) {

int wt[n], tat[n], total\_wt = 0, total\_tat = 0, time = 0, remaining\_bt[n]; for(int i = 0; i < n; i++) remaining\_bt[i] = bt[i]; while(1) { int done = 1;

for(int i = 0; i < n; i++) { if(remaining\_bt[i] > 0) {

done = 0;

if(remaining\_bt[i] > quantum) { time += quantum;

remaining\_bt[i] -= quantum;

} else { time += remaining\_bt[i]; wt[i] = time - bt[i];

remaining\_bt[i] = 0;

}

}

}

if(done == 1) break;

}

for(int i = 0; i < n; i++) { tat[i] = bt[i] + wt[i]; total\_wt += wt[i]; total\_tat += tat[i];

}

printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time"); for(int i = 0; i < n; i++) {

printf("\nP%d\t\t%d\t\t%d\t\t%d", i+1, bt[i], wt[i], tat[i]);

}

float avg\_wt = (float) total\_wt / n; float avg\_tat = (float) total\_tat / n;

printf("\n\nAverage Waiting Time: %f", avg\_wt); printf("\nAverage Turnaround Time: %f\n", avg\_tat);

} int main() { int n = 3, quantum = 4; int bt[] = {24, 3, 3}; round\_robin(bt, n, quantum); }

Output:

