#include <stdio.h>

void round\_robin(int processes[], int n, int burst\_time[], int arrival\_time[], int time\_quantum)

{

int remaining\_time[n];

for (int i = 0; i < n; i++)

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

int current\_time = 0;

while (1)

{

int all\_completed = 1;

for (int i = 0; i < n; i++)

{

if (arrival\_time[i] <= current\_time && remaining\_time[i] > 0)

{

all\_completed = 0;

if (remaining\_time[i] > time\_quantum)

{

current\_time += time\_quantum;

remaining\_time[i] -= time\_quantum;

printf("Executing process %d at time %d\n", processes[i], current\_time);

}

else

{

current\_time += remaining\_time[i];

remaining\_time[i] = 0;

printf("Executing process %d at time %d\n", processes[i], current\_time);

}

}

}

if (all\_completed)

break;

// If no process is available at current time, move to the next time slot

if (all\_completed && current\_time < arrival\_time[n - 1])

current\_time = arrival\_time[n - 1];

}

}

int main()

{

int n;

printf("Enter the number of processes: ");

scanf("%d", &n);

int processes[n];

int burst\_time[n];

int arrival\_time[n];

int time\_quantum;

printf("Enter the burst time and arrival time for each process:\n");

for (int i = 0; i < n; i++)

{

printf("Process %d:\n", i + 1);

printf("Burst Time: ");

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

printf("Arrival Time: ");

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

processes[i] = i + 1;

}

printf("Enter the time quantum: ");

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

round\_robin(processes, n, burst\_time, arrival\_time, time\_quantum);

return 0;

}