#include <stdio.h>

// Structure to represent a process

struct Process {

int pid; // Process ID

int burstTime; // Burst time of the process

int remainingTime; // Remaining burst time of the process

};

// Function to simulate Round Robin scheduling

void roundRobin(struct Process processes[], int n, int quantum) {

int currentTime = 0; // Current time

int completedProcesses = 0; // Number of completed processes

// Loop until all processes are completed

while (completedProcesses < n) {

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

if (processes[i].remainingTime > 0) {

// Execute the process for the quantum or its remaining time, whichever is smaller

int executeTime = (processes[i].remainingTime < quantum) ? processes[i].remainingTime : quantum;

// Update remaining time and current time

processes[i].remainingTime -= executeTime;

currentTime += executeTime;

// Print process execution details

printf("Time %d: Process %d executed for %d units\n", currentTime, processes[i].pid, executeTime);

// Check if the process is completed

if (processes[i].remainingTime == 0) {

completedProcesses++;

}

}

}

}

// Calculate and print average turnaround time

int totalTurnaroundTime = 0;

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

totalTurnaroundTime += (currentTime - processes[i].burstTime);

}

float avgTurnaroundTime = (float)totalTurnaroundTime / n;

printf("\nAverage Turnaround Time: %.2f\n", avgTurnaroundTime);

}

int main() {

// Number of processes

int n;

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

scanf("%d", &n);

// Quantum time

int quantum;

printf("Enter the time quantum: ");

scanf("%d", &quantum);

// Array of processes

struct Process processes[n];

// Input burst times for each process

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

processes[i].pid = i + 1;

printf("Enter burst time for process %d: ", processes[i].pid);

scanf("%d", &processes[i].burstTime);

processes[i].remainingTime = processes[i].burstTime;

}

// Run Round Robin scheduling

roundRobin(processes, n, quantum);

return 0;

}

Output:-

Enter the number of processes: 4

Enter the time quantum: 2

Enter burst time for process 1: 5

Enter burst time for process 2: 3

Enter burst time for process 3: 8

Enter burst time for process 4: 6

Time 2: Process 1 executed for 2 units

Time 4: Process 2 executed for 2 units

Time 6: Process 3 executed for 2 units

Time 8: Process 4 executed for 2 units

Time 10: Process 1 executed for 2 units

Time 12: Process 3 executed for 2 units

Time 14: Process 4 executed for 2 units

Time 16: Process 1 executed for 2 units

Time 18: Process 3 executed for 2 units

Time 20: Process 4 executed for 2 units

Time 22: Process 1 executed for 1 units

Time 23: Process 3 executed for 1 units

Time 24: Process 4 executed for 1 units

Average Turnaround Time: 9.75