**Assignment 4.3**

**Name** : Komal Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

**Round Robin**

|  |
| --- |
| #include <stdio.h>  struct Process {      int id;      int arrival\_time;      int burst\_time;      int remaining\_time;      int turnaround\_time;      int completion\_time;      int start\_time;  };  int main() {      int num\_processes, quantum;      printf("Enter the number of processes: ");      scanf("%d", &num\_processes);      printf("Enter the time quantum: ");      scanf("%d", &quantum);      struct Process processes[num\_processes];      printf("\nEnter arrival time and burst time for each process:\n");      for (int i = 0; i < num\_processes; i++) {          printf("Process %d:\n", i + 1);          printf("Arrival time: ");          scanf("%d", &processes[i].arrival\_time);          printf("Burst time: ");          scanf("%d", &processes[i].burst\_time);          processes[i].id = i + 1;          processes[i].remaining\_time = processes[i].burst\_time;      }      int current\_time = 0;      while (1) {          int all\_finished = 1;          for (int i = 0; i < num\_processes; i++) {              if (processes[i].remaining\_time > 0) {                  all\_finished = 0;                  processes[i].start\_time = current\_time;                  if (processes[i].remaining\_time > quantum) {                      current\_time += quantum;                      processes[i].remaining\_time -= quantum;                  } else {                      current\_time += processes[i].remaining\_time;                      processes[i].remaining\_time = 0;                      processes[i].completion\_time = current\_time;                      processes[i].turnaround\_time = processes[i].completion\_time - processes[i].arrival\_time;                  }              }          }          if (all\_finished)              break;      }      printf("\nProcess\tArrival Time\tBurst Time\tStart Time\tCompletion Time\tTurnaround Time\n");      for (int i = 0; i < num\_processes; i++) {          printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t\t%d\n", processes[i].id, processes[i].arrival\_time,                 processes[i].burst\_time, processes[i].start\_time,                 processes[i].completion\_time, processes[i].turnaround\_time);      }      return 0;  } |

**Output:**

