**Assignment 4.4**

**Name** : Komal Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

**Priority Scheduling**

|  |
| --- |
| #include <stdio.h>  struct Process {      int id;      int arrival\_time;      int burst\_time;      int priority;      int turnaround\_time;      int completion\_time;      int start\_time;  };  void swap(struct Process \*a, struct Process \*b) {      struct Process temp = \*a;      \*a = \*b;      \*b = temp;  }  void sortByPriority(struct Process processes[], int num\_processes) {      for (int i = 0; i < num\_processes - 1; i++) {          for (int j = 0; j < num\_processes - i - 1; j++) {              if (processes[j].priority > processes[j + 1].priority) {                  swap(&processes[j], &processes[j + 1]);              }          }      }  }  int main() {      int num\_processes;      printf("Enter the number of processes: ");      scanf("%d", &num\_processes);      struct Process processes[num\_processes];      printf("\nEnter arrival time, burst time, and priority 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);          printf("Priority: ");          scanf("%d", &processes[i].priority);          processes[i].id = i + 1;      }      sortByPriority(processes, num\_processes);      int current\_time = 0;      for (int i = 0; i < num\_processes; i++) {          processes[i].start\_time = current\_time;          current\_time += processes[i].burst\_time;          processes[i].completion\_time = current\_time;          processes[i].turnaround\_time = processes[i].completion\_time - processes[i].arrival\_time;      }      printf("\nProcess\tArrival Time\tBurst Time\tPriority\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%d\t\t\t%d\n", processes[i].id, processes[i].arrival\_time,                 processes[i].burst\_time, processes[i].priority, processes[i].start\_time,                 processes[i].completion\_time, processes[i].turnaround\_time);      }      return 0;  } |

**Output:**

