**Name-Ujjawal Mandhani**

**Batch-F6**

**Enroll No.-9918103237**

**OSSP LAB-4**

**Question1:**Write a C program for implementing the priority scheduling using Linux operating

system.

**#include <stdio.h>**

**int main()**

**{**

**int burst\_time[20], process[20], waiting\_time[20], turnaround\_time[20], priority[20];**

**int i, j, limit, sum = 0, position, temp;**

**float average\_wait\_time, average\_turnaround\_time;**

**printf("Enter Total Number of Processes:\t");**

**scanf("%d", &limit);**

**printf("\nEnter Burst Time and Priority For %d Processes\n", limit);**

**for (i = 0; i < limit; i++)**

**{**

**printf("\nProcess[%d]\n", i + 1);**

**printf("Process Burst Time:\t");**

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

**printf("Process Priority:\t");**

**scanf("%d", &priority[i]);**

**process[i] = i + 1;**

**}**

**for (i = 0; i < limit; i++)**

**{**

**position = i;**

**for (j = i + 1; j < limit; j++)**

**{**

**if (priority[j] < priority[position])**

**{**

**position = j;**

**}**

**}**

**temp = priority[i];**

**priority[i] = priority[position];**

**priority[position] = temp;**

**temp = burst\_time[i];**

**burst\_time[i] = burst\_time[position];**

**burst\_time[position] = temp;**

**temp = process[i];**

**process[i] = process[position];**

**process[position] = temp;**

**}**

**waiting\_time[0] = 0;**

**for (i = 1; i < limit; i++)**

**{**

**waiting\_time[i] = 0;**

**for (j = 0; j < i; j++)**

**{**

**waiting\_time[i] = waiting\_time[i] + burst\_time[j];**

**}**

**sum = sum + waiting\_time[i];**

**}**

**average\_wait\_time = sum / limit;**

**sum = 0;**

**printf("\nProcess ID\t\tBurst Time\t Waiting Time\t Turnaround Time\n");**

**for (i = 0; i < limit; i++)**

**{**

**turnaround\_time[i] = burst\_time[i] + waiting\_time[i];**

**sum = sum + turnaround\_time[i];**

**printf("\nProcess[%d]\t\t%d\t\t %d\t\t %d\n", process[i], burst\_time[i], waiting\_time[i], turnaround\_time[i]);**

**}**

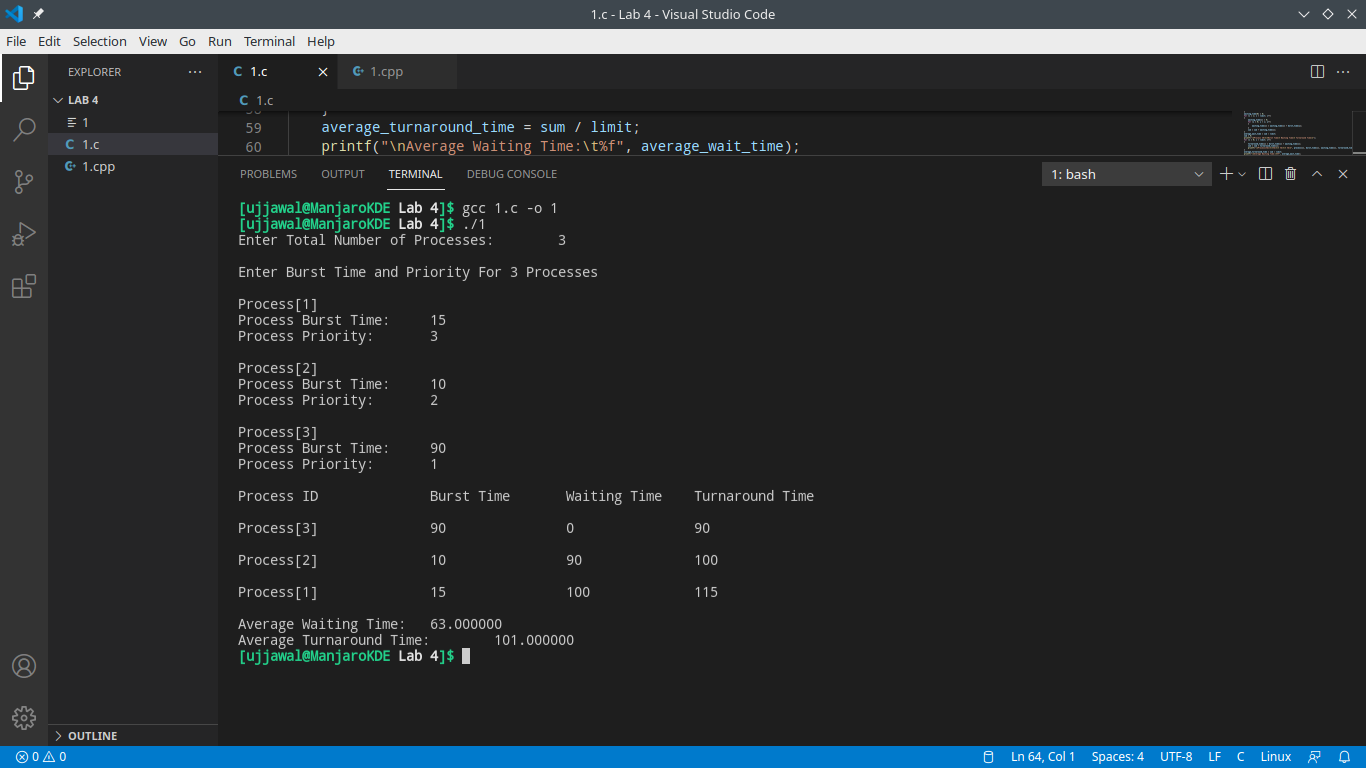
**average\_turnaround\_time = sum / limit;**

**printf("\nAverage Waiting Time:\t%f", average\_wait\_time);**

**printf("\nAverage Turnaround Time:\t%f\n", average\_turnaround\_time);**

**return 0;**

**}**



**Question 2:**Write a Linux C Program for the Implementation of Shortest Job First Scheduling

Algorithm.

**#include <stdio.h>**

**int main()**

**{**

**int bt[20], p[20], wt[20], tat[20], i, j, n, total = 0, pos, temp;**

**float avg\_wt, avg\_tat;**

**printf("Enter number of process:");**

**scanf("%d", &n);**

**printf("\n Enter Burst Time:n");**

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

**{**

**printf("p%d:", i + 1);**

**scanf("%d", &bt[i]);**

**p[i] = i + 1;**

**}**

**//sorting of burst times**

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

**{**

**pos = i;**

**for (j = i + 1; j < n; j++)**

**{**

**if (bt[j] < bt[pos])**

**pos = j;**

**}**

**temp = bt[i];**

**bt[i] = bt[pos];**

**bt[pos] = temp;**

**temp = p[i];**

**p[i] = p[pos];**

**p[pos] = temp;**

**}**

**wt[0] = 0;**

**for (i = 1; i < n; i++)**

**{**

**wt[i] = 0;**

**for (j = 0; j < i; j++)**

**wt[i] += bt[j];**

**total += wt[i];**

**}**

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

**total = 0;**

**printf("\n Processt Burst Time tWaiting TimetTurnaround Time");**

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

**{**

**tat[i] = bt[i] + wt[i];**

**total += tat[i];**

**printf("\n p%dtt %dtt %dttt%d", p[i], bt[i], wt[i], tat[i]);**

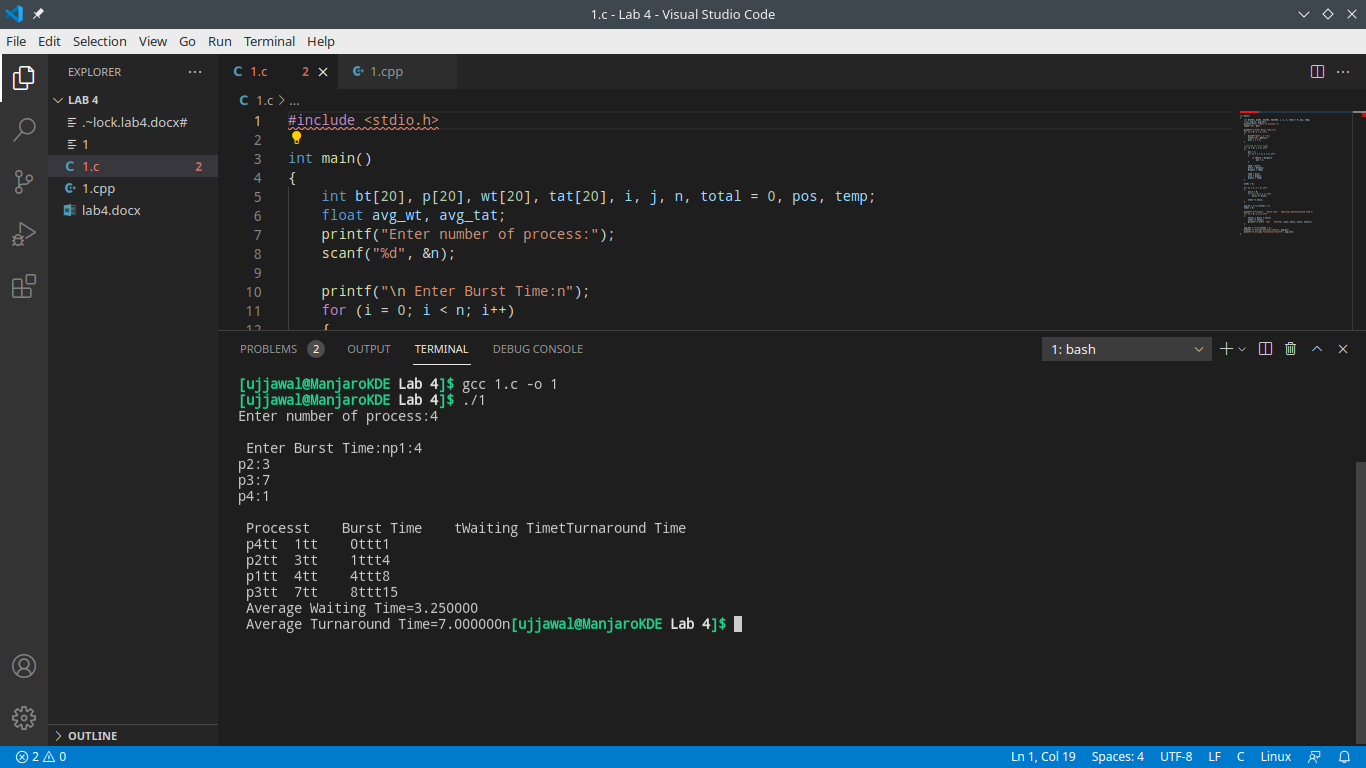
**}**

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

**printf("\n Average Waiting Time=%f", avg\_wt);**

**printf("\n Average Turnaround Time=%fn", avg\_tat);**

**}**



**Question 3:**Write a Linux C Program for the Implementation of Round Robin Scheduling Algorithm.

**#include <stdio.h>**

**#include <stdlib.h>**

**void main()**

**{**

**// initlialize the variable name**

**int i, NOP, sum = 0, count = 0, y, quant, wt = 0, tat = 0, at[10], bt[10], temp[10];**

**float avg\_wt, avg\_tat;**

**printf(" Total number of process in the system: ");**

**scanf("%d", &NOP);**

**y = NOP; // Assign the number of process to variable y**

**// Use for loop to enter the details of the process like Arrival time and the Burst Time**

**for (i = 0; i < NOP; i++)**

**{**

**printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i + 1);**

**printf(" Arrival time is: \t"); // Accept arrival time**

**scanf("%d", &at[i]);**

**printf(" \nBurst time is: \t"); // Accept the Burst time**

**scanf("%d", &bt[i]);**

**temp[i] = bt[i]; // store the burst time in temp array**

**}**

**// Accept the Time qunat**

**printf("Enter the Time Quantum for the process: \t");**

**scanf("%d", &quant);**

**// Display the process No, burst time, Turn Around Time and the waiting time**

**printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");**

**for (sum = 0, i = 0; y != 0;)**

**{**

**if (temp[i] <= quant && temp[i] > 0) // define the conditions**

**{**

**sum = sum + temp[i];**

**temp[i] = 0;**

**count = 1;**

**}**

**else if (temp[i] > 0)**

**{**

**temp[i] = temp[i] - quant;**

**sum = sum + quant;**

**}**

**if (temp[i] == 0 && count == 1)**

**{**

**y--; //decrement the process no.**

**printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\t\t %d", i + 1, bt[i], sum - at[i], sum - at[i] - bt[i]);**

**wt = wt + sum - at[i] - bt[i];**

**tat = tat + sum - at[i];**

**count = 0;**

**}**

**if (i == NOP - 1)**

**{**

**i = 0;**

**}**

**else if (at[i + 1] <= sum)**

**{**

**i++;**

**}**

**else**

**{**

**i = 0;**

**}**

**}**

**// represents the average waiting time and Turn Around time**

**avg\_wt = wt \* 1.0 / NOP;**

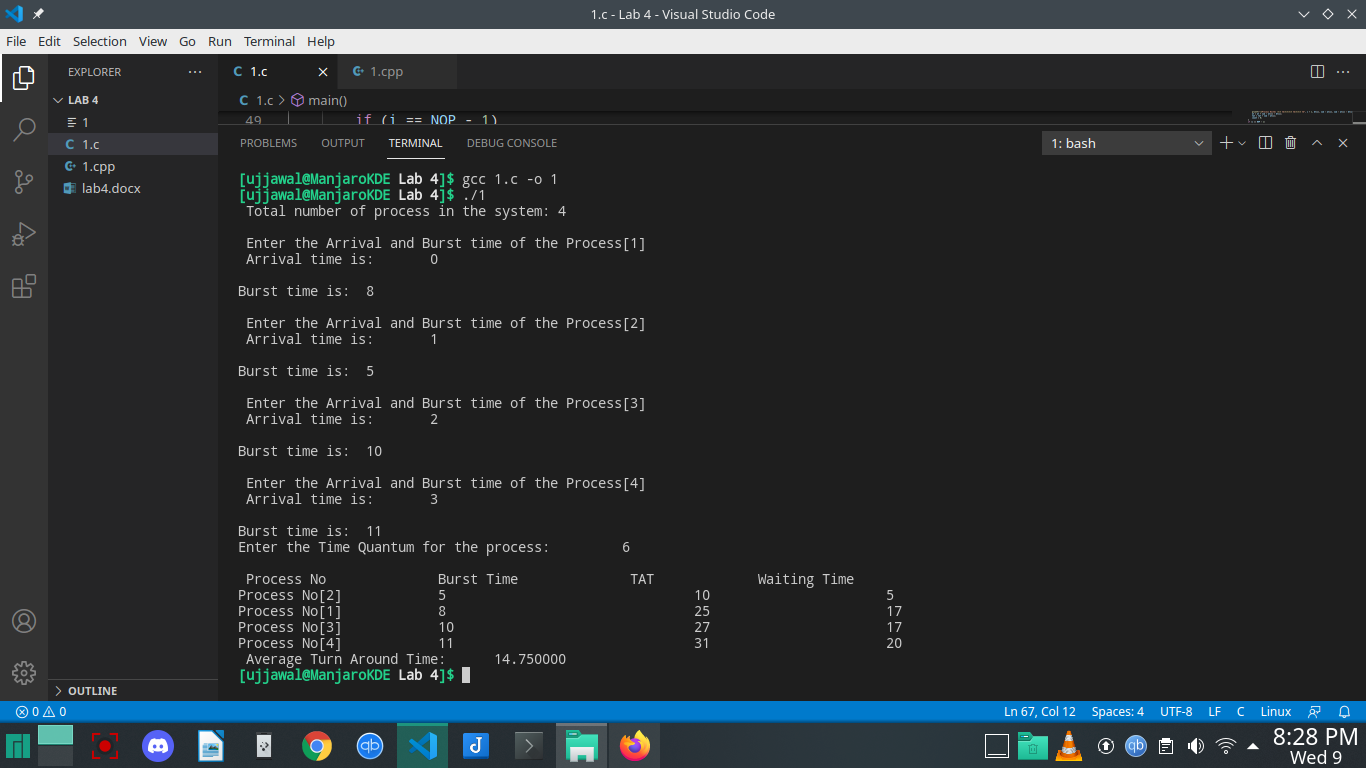
**avg\_tat = tat \* 1.0 / NOP;**

**printf("\n Average Turn Around Time: \t%f", avg\_wt);**

**printf("\n Average Waiting Time: \t%f", avg\_tat);**

**return;**

**}**



**Question 4:**Write a Linux C Program for the Implementation of Multilevel feedback queue

scheduling algorithm. Use FCFS, Shortest Job first and Round Robin as scheduling queue.

**#include <stdio.h>**

**#define N 10**

**typedef struct**

**{**

**int process\_id, arrival\_time, burst\_time, priority;**

**int q, ready;**

**} process\_structure;**

**int Queue(int t1)**

**{**

**if (t1 == 0 || t1 == 1 || t1 == 2 || t1 == 3)**

**{**

**return 1;**

**}**

**else**

**{**

**return 2;**

**}**

**}**

**int main()**

**{**

**int limit, count, temp\_process, time, j, y;**

**process\_structure temp;**

**printf("Enter Total Number of Processes:\t");**

**scanf("%d", &limit);**

**process\_structure process[limit];**

**for (count = 0; count < limit; count++)**

**{**

**printf("\nProcess ID:\t");**

**scanf("%d", &process[count].process\_id);**

**printf("Arrival Time:\t");**

**scanf("%d", &process[count].arrival\_time);**

**printf("Burst Time:\t");**

**scanf("%d", &process[count].burst\_time);**

**printf("Process Priority:\t");**

**scanf("%d", &process[count].priority);**

**temp\_process = process[count].priority;**

**process[count].q = Queue(temp\_process);**

**process[count].ready = 0;**

**}**

**time = process[0].burst\_time;**

**for (y = 0; y < limit; y++)**

**{**

**for (count = y; count < limit; count++)**

**{**

**if (process[count].arrival\_time < time)**

**{**

**process[count].ready = 1;**

**}**

**}**

**for (count = y; count < limit - 1; count++)**

**{**

**for (j = count + 1; j < limit; j++)**

**{**

**if (process[count].ready == 1 && process[j].ready == 1)**

**{**

**if (process[count].q == 2 && process[j].q == 1)**

**{**

**temp = process[count];**

**process[count] = process[j];**

**process[j] = temp;**

**}**

**}**

**}**

**}**

**for (count = y; count < limit - 1; count++)**

**{**

**for (j = count + 1; j < limit; j++)**

**{**

**if (process[count].ready == 1 && process[j].ready == 1)**

**{**

**if (process[count].q == 1 && process[j].q == 1)**

**{**

**if (process[count].burst\_time > process[j].burst\_time)**

**{**

**temp = process[count];**

**process[count] = process[j];**

**process[j] = temp;**

**}**

**else**

**{**

**break;**

**}**

**}**

**}**

**}**

**}**

**printf("\nProcess[%d]:\tTime:\t%d To %d\n", process[y].process\_id, time, time + process[y].burst\_time);**

**time = time + process[y].burst\_time;**

**for (count = y; count < limit; count++)**

**{**

**if (process[count].ready == 1)**

**{**

**process[count].ready = 0;**

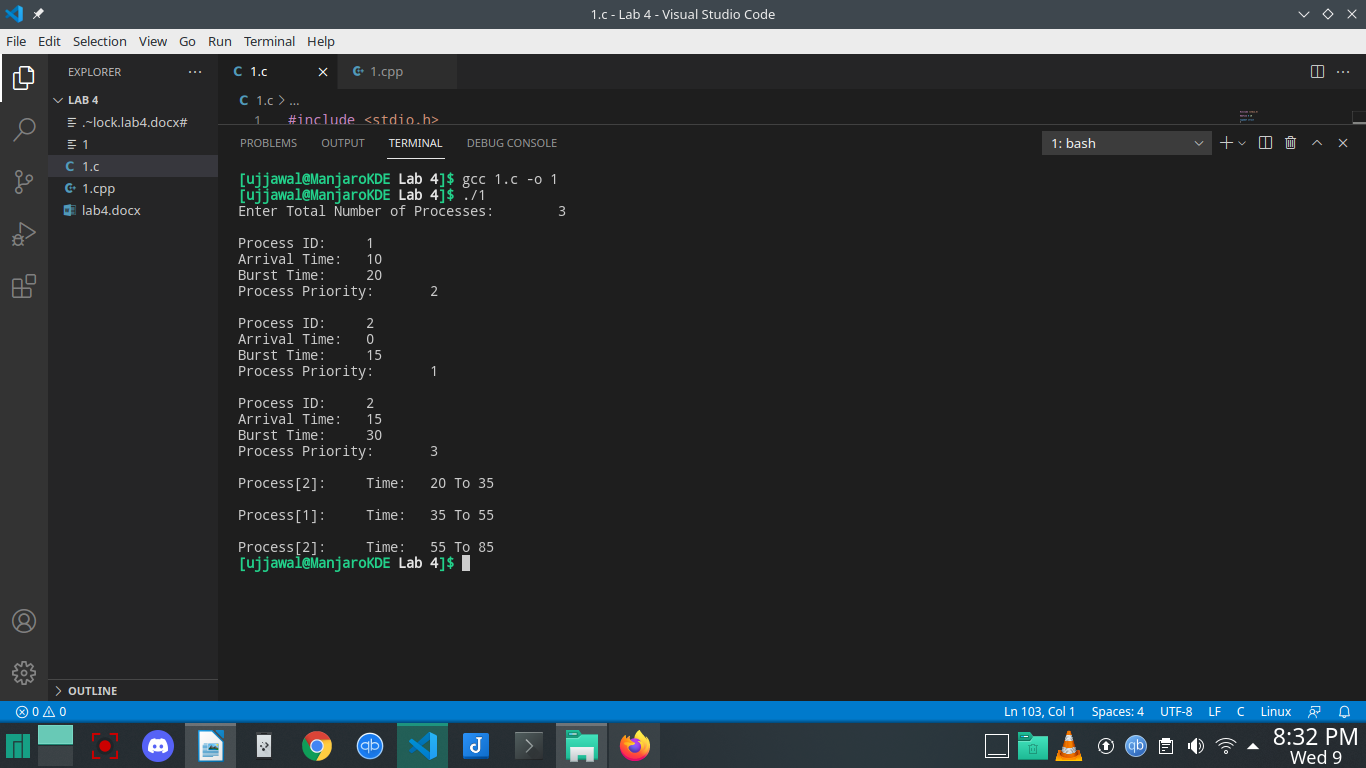
**}**

**}**

**}**

**return 0;**

**}**



**Question 5:**For the above scheduling algorithm, write menu driven Linux c program and give

average waiting time and the average turnaround time in sec.

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**int main()**

**{**

**int p[20], bt[20], su[20], wt[20], tat[20], i, k, n, temp;**

**float wtavg, tatavg;**

**printf("Enter the number of processes --- ");**

**scanf("%d", &n);**

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

**{**

**p[i] = i;**

**printf("Enter the Burst Time of Process %d --- ", i);**

**scanf("%d", &bt[i]);**

**printf("System/User Process (0/1) ? --- ");**

**scanf("%d", &su[i]);**

**}**

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

**for (k = i + 1; k < n; k++)**

**if (su[i] > su[k])**

**{**

**temp = p[i];**

**p[i] = p[k];**

**p[k] = temp;**

**temp = bt[i];**

**bt[i] = bt[k];**

**bt[k] = temp;**

**temp = su[i];**

**su[i] = su[k];**

**su[k] = temp;**

**}**

**wtavg = wt[0] = 0;**

**tatavg = tat[0] = bt[0];**

**for (i = 1; i < n; i++)**

**{**

**wt[i] = wt[i - 1] + bt[i - 1];**

**tat[i] = tat[i - 1] + bt[i];**

**wtavg = wtavg + wt[i];**

**tatavg = tatavg + tat[i];**

**}**

**printf("\nPROCESS\t\t SYSTEM/USER PROCESS \tBURST TIME\tWAITING TIME\tTURNAROUND TIME");**

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

**printf("\n%d \t\t %d \t\t %d \t\t %d \t\t %d ", p[i], su[i], bt[i], wt[i], tat[i]);**

**printf("\nAverage Waiting Time is --- %f", wtavg / n);**

**printf("\nAverage Turnaround Time is --- %f", tatavg / n);**

**return 0;**

**}**

