CS266 ASSIGNMENT 5

NAME:

ARCHIT AGRAWAL

ROLL NO. :

202051213

SECTION:

2

***First Come First Serve***

***Code***

#include<stdio.h>

void arrangeArrival(int num, int mat[][6]){

    for (int i = 0; i < num; i++) {

        for (int j = 0; j < num - i - 1; j++) {

            if (mat[j][1] > mat[j + 1][1]) {

                for (int k = 0; k < 5; k++) {

                    int temp = mat[j][k];

                    mat[j][k] = mat[j + 1][k];

                    mat[j + 1][k] = temp;

                }

            }

        }

    }

}

void completionTime(int num, int mat[][6]){

    mat[0][3] = mat[0][2];

    for(int i = 1; i < num; i++){

        if(mat[i - 1][3] < mat[i][1]){

            mat[i][3] = mat[i][1] + mat[i][2];

        } else {

            mat[i][3] = mat[i - 1][3] + mat[i][2];

        }

    }

    for(int i = 0; i < num; i++){

        mat[i][5] = mat[i][3] - mat[i][1];

    }

    for(int i = 0; i < num; i++){

        mat[i][4] = mat[i][5] - mat[i][2];

    }

}

double avgWaitTime(int num, int mat[][6]){

    int totalWaitTime = 0;

    for(int i = 0; i < num; i++){

        totalWaitTime += mat[i][4];

    }

    return (double) totalWaitTime/num;

}

double avgTATime(int num, int mat[][6]){

    int totalTATime = 0;

    for(int i = 0; i < num; i++){

        totalTATime += mat[i][5];

    }

    return (double) totalTATime/num;

}

int main(){

   int n;

   printf("Enter the number of processes ");

   scanf("%d", &n);

   int mat[n][6];

   for(int i = 0; i < n; i++){

       printf("Enter Process ID ");

       int id, at, bt;

       scanf("%d", &id);

       mat[i][0] = id;

       printf("Enter Arrival Time ");

       scanf("%d", &at);

       mat[i][1] = at;

       printf("Enter Burst Time ");

       scanf("%d", &bt);

       mat[i][2] = bt;

    }

    arrangeArrival(n, mat);

    completionTime(n, mat);

    printf("\t\t\t\tFirst Come First Serve\n");

    printf("Process ID \t Arrival Time \t Burst Time \t Completion Time \t Turn Around Time \t Waiting Time\n");

    for(int i = 0; i < n; i++){

        printf("    %2d\t\t\t%d\t     %d\t\t\t%d\t\t\t%d\t\t\t%d\n", mat[i][0], mat[i][1], mat[i][2], mat[i][3], mat[i][5], mat[i][4]);

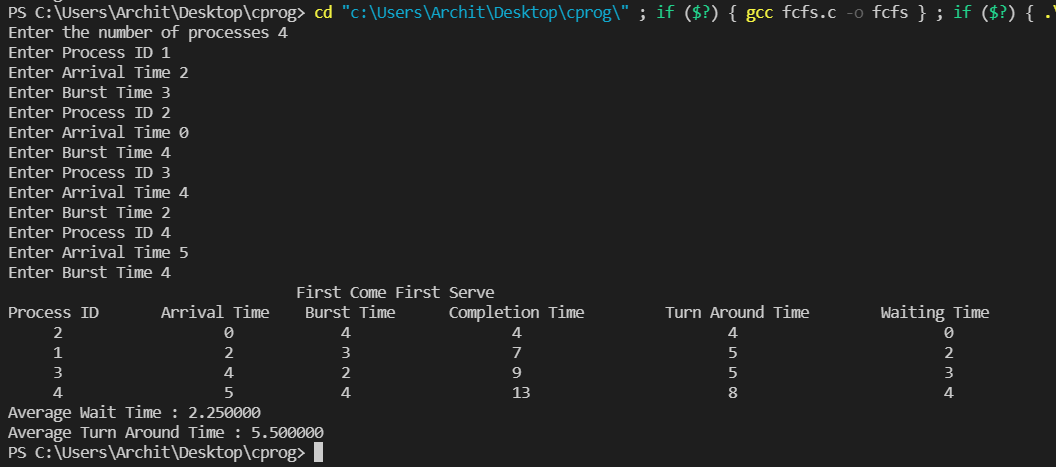
    }

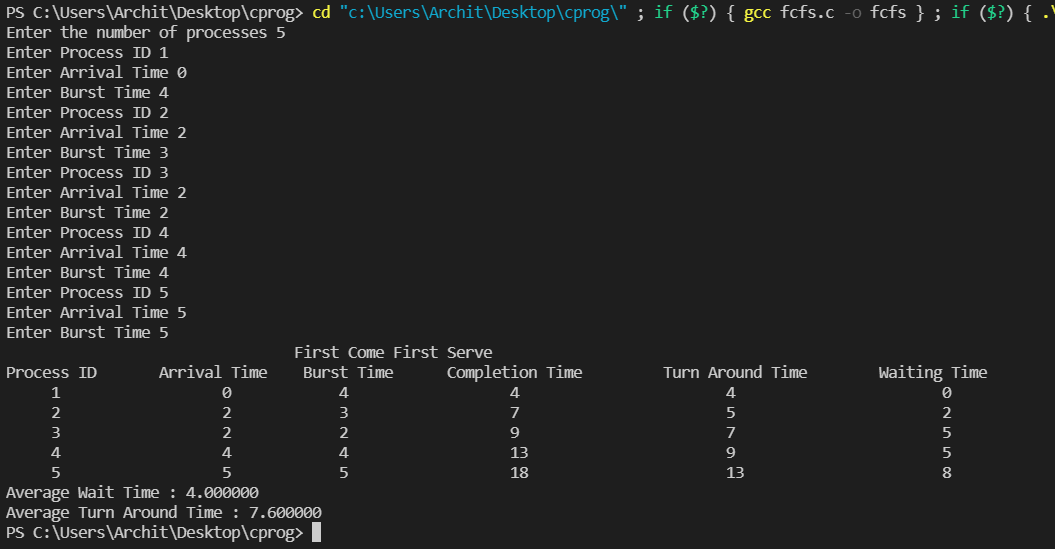
    printf("Average Wait Time : %f\n", avgWaitTime(n, mat));

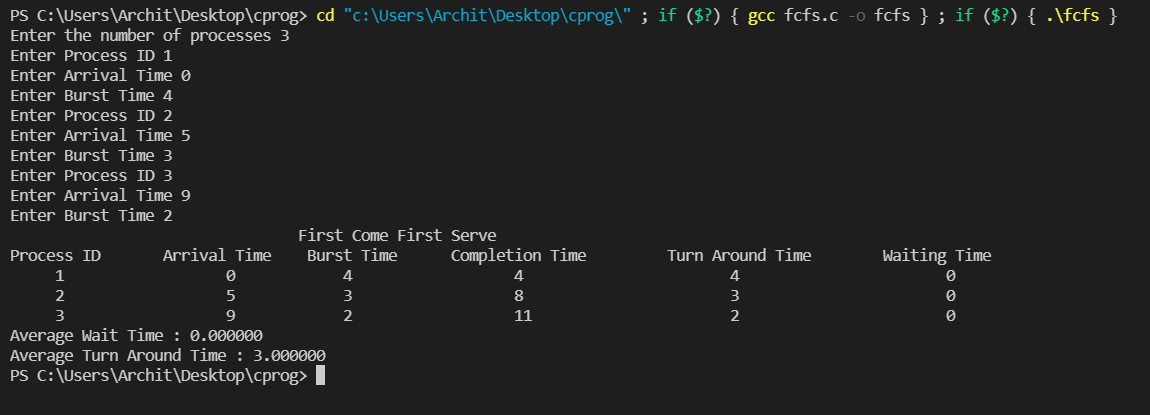
    printf("Average Turn Around Time : %f", avgTATime(n, mat));

}

***Output***







***Shortest Job First***

***Code***

#include<stdio.h>

void arrangeArrival(int num, int mat[][6]){

    for (int i = 0; i < num; i++) {

        for (int j = 0; j < num - i - 1; j++) {

            if (mat[j][1] > mat[j + 1][1]) {

                for (int k = 0; k < 5; k++) {

                    int temp = mat[j][k];

                    mat[j][k] = mat[j + 1][k];

                    mat[j + 1][k] = temp;

                }

            }

        }

    }

}

void completionTime(int num, int mat[][6]){

    int temp, val;

    mat[0][3] = mat[0][1] + mat[0][2];

    mat[0][5] = mat[0][3] - mat[0][1];

    mat[0][4] = mat[0][5] - mat[0][2];

    for (int i = 1; i < num; i++) {

        temp = mat[i - 1][3];

        //in case if CPU is idle

        //we will neglect that time

        if(temp < mat[i][1]){

            temp = mat[i][1];

        }

        int low = mat[i][2];

        for (int j = i; j < num; j++) {

            if (temp >= mat[j][1] && low >= mat[j][2]) {

                low = mat[j][2];

                val = j;

            }

        }

        mat[val][3] = temp + mat[val][2];

        mat[val][5] = mat[val][3] - mat[val][1];

        mat[val][4] = mat[val][5] - mat[val][2];

        for (int k = 0; k < 6; k++) {

            int temp1 = mat[val][k];

            mat[val][k] = mat[i][k];

            mat[i][k] = temp1;

        }

    }

}

double avgWaitTime(int num, int mat[][6]){

    int totalWaitTime = 0;

    for(int i = 0; i < num; i++){

        totalWaitTime += mat[i][4];

    }

    return (double) totalWaitTime/num;

}

double avgTATime(int num, int mat[][6]){

    int totalTATime = 0;

    for(int i = 0; i < num; i++){

        totalTATime += mat[i][5];

    }

    return (double) totalTATime/num;

}

int main(){

   int n;

   printf("Enter the number of processes ");

   scanf("%d", &n);

   int mat[n][6];

   for(int i = 0; i < n; i++){

       printf("Enter Process ID ");

       int id, at, bt;

       scanf("%d", &id);

       mat[i][0] = id;

       printf("Enter Arrival Time ");

       scanf("%d", &at);

       mat[i][1] = at;

       printf("Enter Burst Time ");

       scanf("%d", &bt);

       mat[i][2] = bt;

    }

    arrangeArrival(n, mat);

    completionTime(n, mat);

    printf("\t\t\t\tShortest Job First\n");

    printf("Process ID \t Arrival Time \t Burst Time \t Completion Time \t Turn Around Time \t Waiting Time\n");

    for(int i = 0; i < n; i++){

        printf("    %2d\t\t\t%d\t     %d\t\t\t%d\t\t\t%d\t\t\t%d\n", mat[i][0], mat[i][1], mat[i][2], mat[i][3], mat[i][5], mat[i][4]);

    }

    printf("Average Wait Time : %f\n", avgWaitTime(n, mat));

    printf("Average Turn Around Time : %f", avgTATime(n, mat));

}

***Output***

