Write a program to implement Shortest Job First Scheduling Algorithm. Calculate average waiting time, average turnaround time.( Given the list of processes)

#include <stdio.h>

#define max 30

int main()

{

    int i, j, n, temp, p[max], bt[max], wt[max], tat[max];

    float avg\_wt = 0, avg\_tat = 0;

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

    scanf("%d", &n);

    printf("Enter the  process number : ");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &p[i]);

    }

    printf("Enter the burst time of the process ");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &bt[i]);

    }

    // bubble sort

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < n - i - 1; j++)

        {

            if (bt[j] > bt[j + 1])

            {

                temp = bt[j];

                bt[j] = bt[j + 1];

                bt[j + 1] = temp;

                temp = p[j];

                p[j] = p[j + 1];

                p[j + 1] = temp;

            }

        }

    }

    printf("Process\tBurst time\tWaiting time\tTurn around time\n");

    for(i=0;i<n;i++){

        wt[i]=0;

        tat[i]=0;

        for(j=0;j<i;j++){

            wt[i] =wt[i]+bt[j];

        }

        tat[i] = wt[i]+bt[i];

        avg\_wt +=wt[i];

        avg\_tat  += tat[i];

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

    }

    avg\_tat = avg\_tat/n;

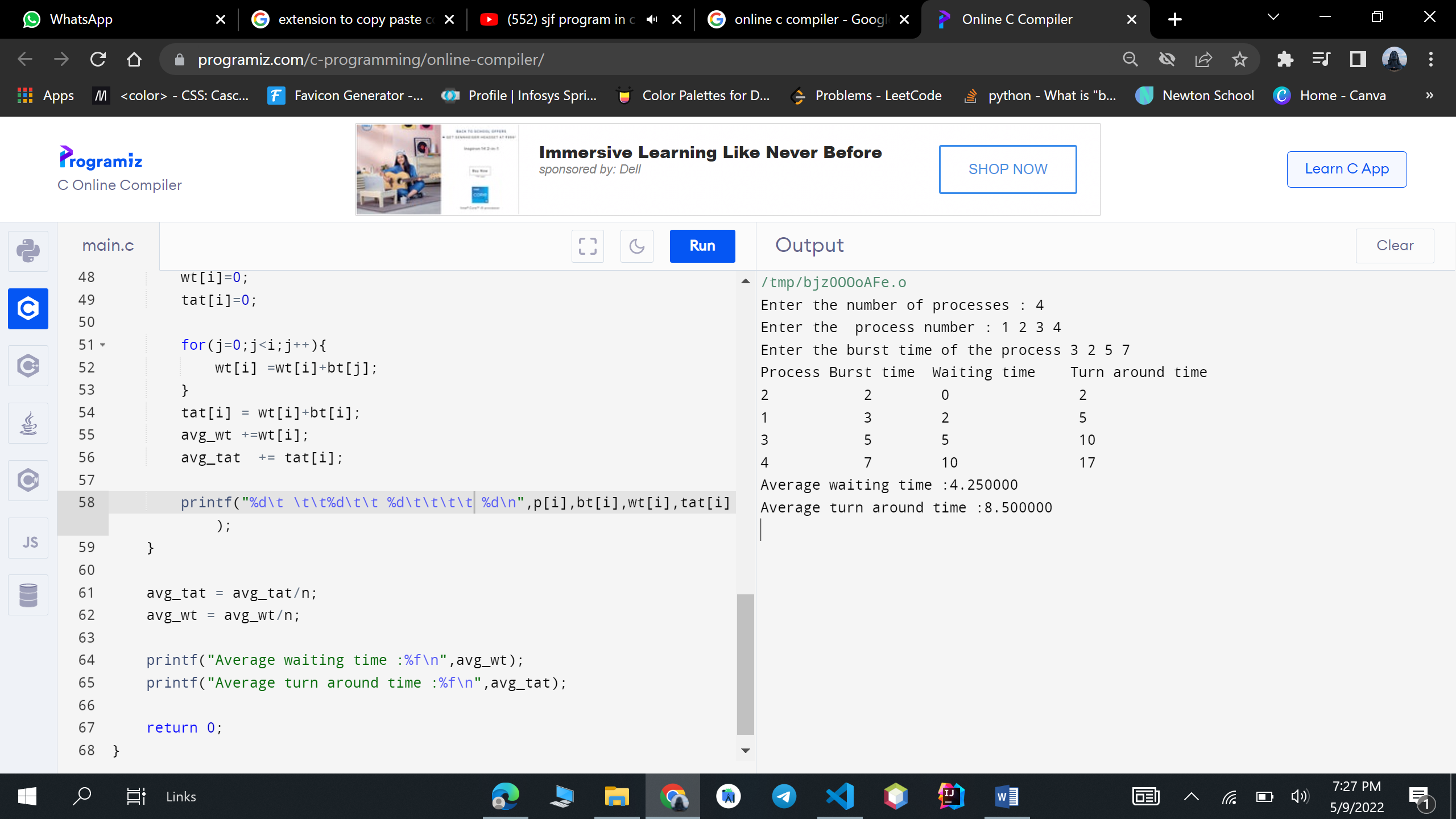
    avg\_wt = avg\_wt/n;

    printf("Average waiting time :%f\n",avg\_wt);

    printf("Average turn around time :%f\n",avg\_tat);

    return 0;

}



Write a program to implement Shortest Remaining Time First Scheduling/ Round Robin (RR)/ Priority Scheduling/ Multilevel Queue Scheduling Algorithm. Algorithm.Calculate average waiting time, average turnaround time and throughput.( Given the list of processes, their CPU burst times and arrival times/ Priorities)

#include <stdio.h>

int main(){

    int n,i,count=0,qt,temp=0,sq=0;

    int bt[10],wt[10],tat[10],rem\_bt[10];

    float awt=0,atat=0;

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

    scanf("%d", &n);

    printf("Enter the burst time of the process : ");

    for (i = 0; i < n; i++)

    {

        scanf("%d", &bt[i]);

        rem\_bt[i]=bt[i];

    }

    printf("Enter the quantum time: ");

    scanf("%d",& qt);

    while (1)

    {

        for(i=0,count=0;i<n;i++){

            temp=qt;

            if(rem\_bt[i]==0){

                count++;

                continue;

            }

            if(rem\_bt[i]>qt){

                rem\_bt[i] =rem\_bt[i]-qt;

            }

            else

            if(rem\_bt[i]>=0){

                temp = rem\_bt[i];

                rem\_bt[i]=0;

            }

            sq+=temp;

            tat[i]=sq;

        }

        if(n==count)

            break;

    }

    printf("\nProcess\tBurst Time\t Turn around time\tWainting time\n");

    for(i=0;i<n;i++){

        wt[i]=tat[i]-bt[i];

        awt += wt[i];

        atat+=tat[i];

        printf("\n%d\t\t%d\t\t\t\t%d\t\t\t\t%d",i+1,bt[i],tat[i],wt[i]);

    }

    awt = awt/n;

    atat = atat/n;

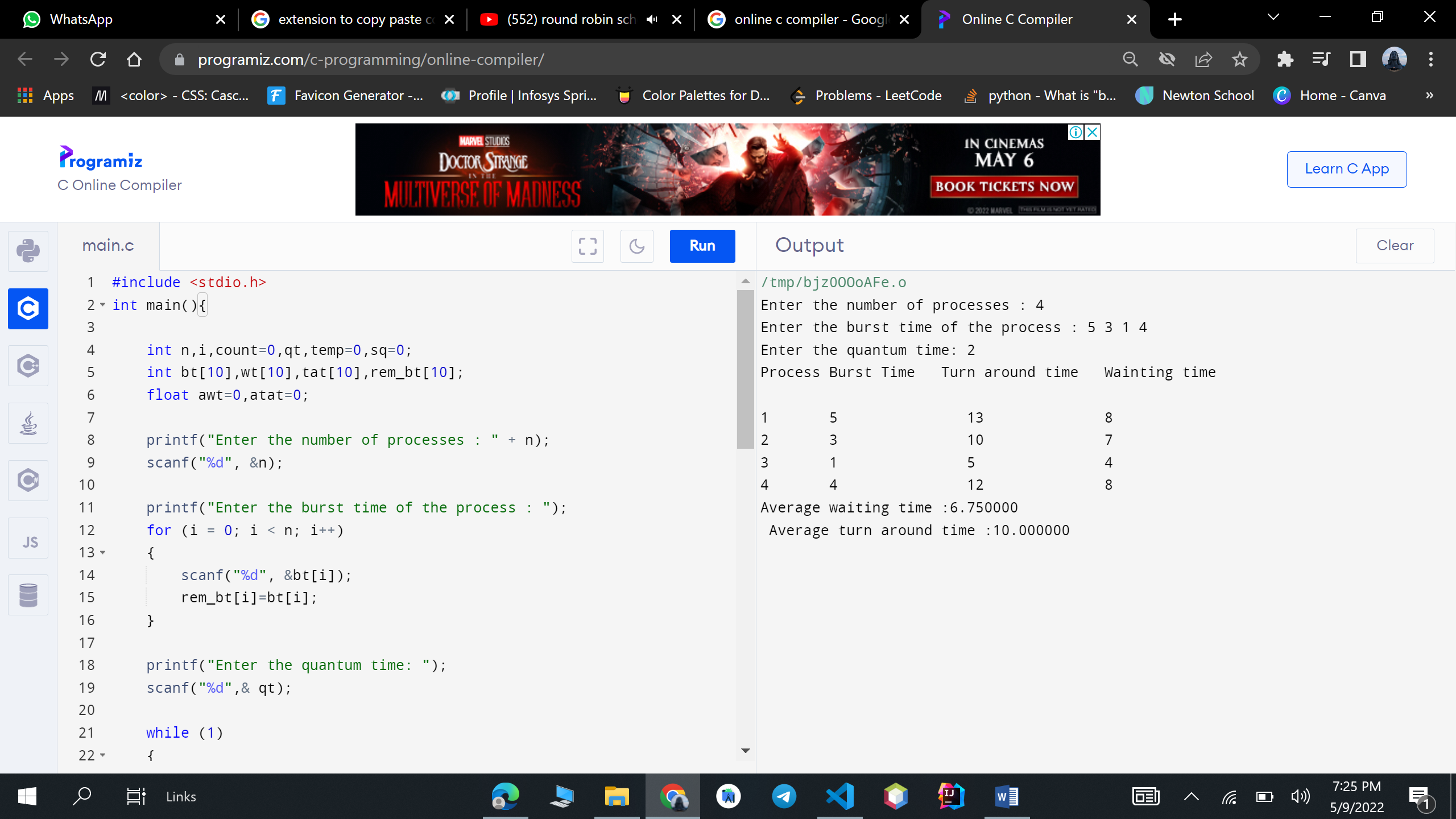
    printf("\nAverage waiting time :%f \n ",awt);

     printf("Average turn around time :%f \n ",atat);

    return 0;

}

**Output:**

****