**Round Robin CPU Scheduling–**

#include <stdio.h>

*int* main()

{

*int* cnt, j, n, t, remain, flag = 0, tq;

*int* wt = 0, tat = 0, at[10], bt[10], rt[10];

    printf("Enter Total Process:\t ");

    scanf("%d", &n);

    remain = n;

    for (cnt = 0; cnt < n; cnt++)

    {

        printf("Enter Arrival Time and Burst Time for Process Process Number %d :", cnt + 1);

        scanf("%d", &at[cnt]);

        scanf("%d", &bt[cnt]);

        rt[cnt] = bt[cnt];

    }

    printf("Enter Time Quantum:\t");

    scanf("%d", &tq);

    printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");

    for (t = 0, cnt = 0; remain != 0;)

    {

        if (rt[cnt] <= tq && rt[cnt] > 0)

        {

            t += rt[cnt];

            rt[cnt] = 0;

            flag = 1;

        }

        else if (rt[cnt] > 0)

        {

            rt[cnt] -= tq;

            t += tq;

        }

        if (rt[cnt] == 0 && flag == 1)

        {

            remain--;

            printf("P[%d]\t|\t%d\t|\t%d\n", cnt + 1, t - at[cnt], t - at[cnt] - bt[cnt]);

            wt += t - at[cnt] - bt[cnt];

            tat += t - at[cnt];

            flag = 0;

        }

        if (cnt == n - 1)

            cnt = 0;

        else if (at[cnt + 1] <= t)

            cnt++;

        else

            cnt = 0;

    }

    printf("\nAverage Waiting Time= %f\n", wt \* 1.0 / n);

    printf("Avg Turnaround Time = %f", tat \* 1.0 / n);

    return 0;

}

**Shortest Job First CPU Scheduling-**

#include <stdio.h>

*int* main()

{

*int* time, burst\_time[10], at[10], sum\_burst\_time = 0, smallest, n, i;

*int* sumt = 0, sumw = 0;

    printf("Enter the no of processes : ");

    scanf("%d", &n);

    for (i = 0; i < n; i++)

    {

        printf("The arrival time for process P%d : ", i + 1);

        scanf("%d", &at[i]);

        printf("The burst time for process P%d : ", i + 1);

        scanf("%d", &burst\_time[i]);

        sum\_burst\_time += burst\_time[i];

    }

    burst\_time[9] = 9999;

    for (time = 0; time < sum\_burst\_time;)

    {

        smallest = 9;

        for (i = 0; i < n; i++)

        {

            if (at[i] <= time && burst\_time[i] > 0 && burst\_time[i] < burst\_time[smallest])

                smallest = i;

        }

        printf("P[%d]\t|\t%d\t|\t%d\n", smallest + 1, time + burst\_time[smallest] - at[smallest], time - at[smallest]);

        sumt += time + burst\_time[smallest] - at[smallest];

        sumw += time - at[smallest];

        time += burst\_time[smallest];

        burst\_time[smallest] = 0;

    }

    printf("\n\n Average waiting time = %f", sumw \* 1.0 / n);

    printf("\n\n Average turnaround time = %f", sumt \* 1.0 / n);

    return 0;

}