

Task 10: Write a program to find the finish time, turnaround time, and waiting time using Priority Scheduling Algorithm (input Takes by user).

```
//priority
#include <stdio.h>

void findWaitingTime(int n, int bt[], int wt[], int pri[]) {
    int rt[n];
    for (int i = 0; i < n; i++) {
        rt[i] = bt[i];
    }

    int complete = 0, t = 0, minm = 9999;
    int shortest = 0, finish_time;
    int flag = 0;

    while (complete != n) {
        for (int j = 0; j < n; j++) {
            if ((pri[j] <= pri[shortest]) && (rt[j] < rt[shortest]) && (rt[j] > 0)) {
                shortest = j;
            }
        }

        rt[shortest]--;

        if (rt[shortest] == 0) {
            complete++;
            flag = 0;
            finish_time = t + 1;
            wt[shortest] = finish_time - bt[shortest];
            if (wt[shortest] < 0) {
```

```

        wt[shortest] = 0;
    }
}
t++;
}
}

```

```

void findTurnAroundTime(int n, int bt[], int wt[], int tat[]) {
    for (int i = 0; i < n; i++) {
        tat[i] = bt[i] + wt[i];
    }
}

```

```

void findFinishTime(int n, int at[], int bt[], int wt[], int ft[]) {
    for (int i = 0; i < n; i++) {
        ft[i] = at[i] + bt[i] + wt[i];
    }
}

```

```

int main() {
    int n;

    printf("Enter the number of processes: ");
    scanf("%d", &n);

    int burst_time[n], arrival_time[n], waiting_time[n], turnaround_time[n], finish_time[n], priority[n];

    for (int i = 0; i < n; i++) {
        printf("Enter arrival time for process %d: ", i + 1);
        scanf("%d", &arrival_time[i]);
        printf("Enter burst time for process %d: ", i + 1);
        scanf("%d", &burst_time[i]);
    }
}

```

```

    printf("Enter priority for process %d: ", i + 1);

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

findWaitingTime(n, burst_time, waiting_time, priority);

findTurnAroundTime(n, burst_time, waiting_time, turnaround_time);

findFinishTime(n, arrival_time, burst_time, waiting_time, finish_time);

printf("\nPID\tArrival Time\tBurst Time\tPriority\tFinish Time\tTurnaround Time\tWaiting
Time\n");

for (int i = 0; i < n; i++) {

    printf("%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n", i + 1, arrival_time[i], burst_time[i],
priority[i], finish_time[i], turnaround_time[i], waiting_time[i]);

}

return 0;
}

```

Output-

```

Enter the number of processes: 2
Enter arrival time for process 1: 0
Enter burst time for process 1: 4
Enter priority for process 1: 2
Enter arrival time for process 2: 1
Enter burst time for process 2: 5
Enter priority for process 2: 1

```

PID	Arrival Time	Burst Time	Priority	Finish Time	Turnaround Time	Waiting Time
1	0	4	2	4	4	0
2	1	5	1	6	5	0