

WITH ARRIVAL TIME --

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

```
void priority_preemptive(int processes[], int n, int burst_time[], int priority[], int arrival_time[])
{
    int remaining_time[n];
    for (int i = 0; i < n; i++)
        remaining_time[i] = burst_time[i];

    int completed = 0, current_time = 0;
    while (completed != n)
    {
        int highest_priority = -1;
        for (int i = 0; i < n; i++)
        {
            if (arrival_time[i] <= current_time && remaining_time[i] > 0)
            {
                if (highest_priority == -1 || priority[i] < priority[highest_priority])
                    highest_priority = i;
            }
        }

        if (highest_priority == -1)
        {
            current_time++;
            continue;
        }

        printf("Executing process %d at time %d\n", processes[highest_priority], current_time);

        remaining_time[highest_priority]--;
        current_time++;

        if (remaining_time[highest_priority] == 0)
        {
            completed++;
            printf("Process %d completed at time %d\n", processes[highest_priority], current_time);
        }
    }
}
```

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

    int processes[n];
    int burst_time[n];
    int priority[n];
    int arrival_time[n];
    printf("Enter the burst time, priority, and arrival time for each process:\n");
    for (int i = 0; i < n; i++)
    {
        printf("Process %d:\n", i + 1);
```

```

        printf("BurstTime: ");
        scanf("%d", &burst_time[i]);
        printf("Priority: ");
        scanf("%d", &priority[i]);
        printf("Arrival Time: ");
        scanf("%d", &arrival_time[i]);
        processes[i] = i + 1;
    }

    priority_preemptive(processes, n, burst_time, priority, arrival_time);

    return 0;
}

```

WITHOUT ARRIVAL TIME

-->

```

#include <stdio.h>

void priority_preemptive(int processes[], int n, int burst_time[], int priority[])
{
    int remaining_time[n];
    for (int i = 0; i < n; i++)
        remaining_time[i] = burst_time[i];

    int completed = 0, current_time = 0;
    while (completed != n)
    {
        int highest_priority = -1;
        for (int i = 0; i < n; i++)
        {
            if (remaining_time[i] > 0)
            {
                if (highest_priority == -1 || priority[i] < priority[highest_priority])
                    highest_priority = i;
            }
        }

        printf("Executing process %d at time %d\n", processes[highest_priority], current_time);

        remaining_time[highest_priority]--;
        current_time++;

        if (remaining_time[highest_priority] == 0)
        {
            completed++;
            printf("Process %d completed at time %d\n", processes[highest_priority], current_time);
        }
    }
}

int main()
{
    int n;

```

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

int processes[n];
int burst_time[n];
int priority[n];
printf("Enter the burst time and priority for each process:\n");
for (int i = 0; i < n; i++)
{
    printf("Process %d:\n", i + 1);
    printf("BurstTime: ");
    scanf("%d", &burst_time[i]);
    printf("Priority:");
    scanf("%d", &priority[i]);
    processes[i] = i + 1;
}

priority_preemptive(processes, n, burst_time, priority);

return 0;
}
```