

```

#include <stdio.h>
void line(int n)
{
    for (int i = 0; i < n; i++)
        printf("=");
    printf("\n");
}

int abs(int a)
{
    if(a < 0)
        return -a;
    return a;
}

int main()
{
    int p;
    printf("Enter number of processes : ");
    scanf("%d", &p);
    p = abs(p);

    int at[p], bt[p], pri[p];

    int total_time = 0;
    printf("Enter arrival and Burst time and Priority\n");
    for (int i = 0; i < p; i++)
    {
        printf("Process %d : ", i + 1);
        scanf("%d", &at[i]);
        scanf("%d", &bt[i]);
        scanf("%d", &pri[i]);

        at[i] = abs(at[i]);
        bt[i] = abs(bt[i]);
        pri[i] = abs(pri[i]);

        total_time += bt[i];
    }
    printf("\n");

    int clock = 0;
    int gantt[total_time];
    int exe = -1;

    while(clock < total_time)
    {
        for(int i=0;i<p;i++)
        {
            if(at[i] <= clock && bt[i] > 0)
            {
                if(exe == -1)
                {
                    exe = i;
                }
                else
                {
                    if(pri[i] < pri[exe] || bt[exe] == 0)
                    {
                        exe = i;
                    }
                }
            }
        }

        gantt[clock] = exe;
        clock++;
        bt[exe]--;
    }

    line(total_time * 2);
    for(int i=0;i<total_time;i++)
    {
        printf("%d ", gantt[i] + 1);
    }
    printf("\n");
    line(total_time * 2);

    int ct[p], tat[p], wt[p], t_bt[p];

    for (int i = 0; i < p; i++)
    {
        int tbt = 0;
        int tstart = -1;
        int last = -1;
        for (int j = 0; j < total_time; j++)
        {

```

```

        if (gantt[j] == i)
        {
            tbt++;
            last = j;
            if (tstart == -1)
                tstart = j;
        }
    }

    t_bt[i] = tbt;
    ct[i] = last + 1;
    tat[i] = ct[i] - at[i];
    wt[i] = tat[i] - t_bt[i];
}

line(77);
printf("%10s|%10s|%10s|%10s|%10s|%10s|%10s|\n", "Process No", "A. T.", "B. T.", "Priority", "C. T.", "T. A. T.",
"W. T.");
for (int i = 0; i < p; i++)
{
    printf("%10d|%10d|%10d|%10d|%10d|%10d|%10d|\n", i+1, at[i], t_bt[i], pri[i], ct[i], tat[i], wt[i]);
}
line(77);

double avg_tat = 0, avg_wt = 0;
for (int i = 0; i < p; i++)
{
    avg_tat += tat[i];
    avg_wt += wt[i];
}

avg_tat = avg_tat / p;
avg_wt = avg_wt / p;

printf("Average Turn Around Time : %f\n", avg_tat);
printf("Average Waiting Time : %f\n", avg_wt);

return 0;
}

```

```

// 5 8 2 3 3 8 5 0 15 4 16 4 1 10 11 2
// 5 5 4 5 0 9 4 15 7 2 12 12 1 8 6 3

```

/*OUTPUT -

```

Enter number of processes : 5
Enter arrival and Burst time and Priority
Process 1 : 8 2 3
Process 2 : 3 8 5
Process 3 : 0 15 4
Process 4 : 16 4 1
Process 5 : 10 11 2

```

```

=====
3 3 3 3 3 3 3 3 1 1 5 5 5 5 5 5 4 4 4 4 5 5 5 5 3 3 3 3 3 3 2 2 2 2 2 2 2
=====

```

Process No	A. T.	B. T.	Priority	C. T.	T. A. T.	W. T.
1	8	2	3	10	2	0
2	3	8	5	40	37	29
3	0	15	4	32	32	17
4	16	4	1	20	4	0
5	10	11	2	25	15	4

```

=====
Average Turn Around Time : 18.000000
Average Waiting Time : 10.000000

```

```

Enter number of processes : 5
Enter arrival and Burst time and Priority
Process 1 : 5 4 5
Process 2 : 0 9 4
Process 3 : 15 7 2
Process 4 : 12 12 1
Process 5 : 8 6 3

```

```

=====
2 2 2 2 2 2 2 2 5 5 5 5 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 5 5 2 1 1 1 1
=====

```

Process No	A. T.	B. T.	Priority	C. T.	T. A. T.	W. T.
1	5	4	5	38	33	29
2	0	9	4	34	34	25
3	15	7	2	31	16	9
4	12	12	1	24	12	0
5	8	6	3	33	25	19

```

=====
Average Turn Around Time : 24.000000
Average Waiting Time : 16.400000

```