PROGRAM-8

```
#include <stdio.h>

struct process {
        int at, bt, ct, wt, tt,id;
        int completed;
        float ntt;
} p[10];

int n;
void sortByArrival()
{
        struct process temp;
        int i, j;
        for (i = 0; i < n - 1; i++) {
                for (j = i + 1; j < n; j++) {
                        if (p[i].at > p[j].at) {
                                temp = p[i];
                                p[i] = p[j];
                                p[j] = temp;
                        }
                }
        }
}

void main()
{
        int i, j, t, sum_bt = 0;
        float avgwt = 0, avgtt = 0;
    printf("number of processes ");
    scanf("%d",&n);
        printf("AT\tBT\n");
        for (i=0;i<n;i++)
    {
                p[i].id=i+1;
                scanf("%d %d",&p[i].at,&p[i].bt);
                p[i].completed = 0;
                sum_bt += p[i].bt;
        }
        sortByArrival();
        printf("\nName\tArrival Time\tBurst Time\tWaiting Time");
        printf("\tTurnAround Time\t Normalized TT");
        for (t = p[0].at; t < sum_bt;) {
                float hrr = -9999;
                float temp;
                int loc;
                for (i = 0; i < n; i++) {
                        if (p[i].at <= t && p[i].completed != 1)
                        {
                                temp = (p[i].bt + (t - p[i].at)) /
p[i].bt;
```

```
                                if (hrr < temp)
                                {
                                        hrr = temp;
                                        loc = i;
                                }
                        }
                }
                t += p[loc].bt;
                p[loc].wt = t - p[loc].at - p[loc].bt;
                p[loc].tt = t - p[loc].at;
                avgtt += p[loc].tt;
                p[loc].ntt = ((float)p[loc].tt / p[loc].bt);
                p[loc].completed = 1;
                avgwt += p[loc].wt;
                printf("\n%c\t\t%d\t\t", p[loc].name, p[loc].at);
                printf("%d\t\t%d\t\t", p[loc].bt, p[loc].wt);
                printf("%d\t\t%f", p[loc].tt, p[loc].ntt);
        }
        printf("\nAverage waiting time:%f\n", avgwt / n);
        printf("Average Turn Around time:%f\n", avgtt / n);
}
```