```
//Shorest Remaining Time First
#include <stdio.h>
#define MAX 100

struct process
{
    int AT;
    int BT;
    int id;
    int CT;
    int WT;
    int TAT;
    int rem;
};

void swap(process &a, process &b)
{
    process t = a;
    a = b;
    b = t;
}

int get_partion(process A[], int start , int end)
{
    int pivot = A[end].AT;
    int i=start-1;
    for(int j=start ; j<end ; j++)
    {
        if(A[j].AT <= pivot)
        {
            i++;
            swap(A[i], A[j]);
        }
    }
    swap(A[end], A[i+1]);
    return i+1;
}

void quicksort(process A[], int start, int end)
{
    if(start < end)
    {
        int partion = get_partion(A, start, end);
        quicksort(A, start, partion-1);
        quicksort(A, partion+1, end);
    }
}

void simulate(process P[], int n)
{
    int count=0;
    for(int time=0; count!=n; time++)
    {
        int smallest = MAX;
        for(int i=0 ; i<n ; i++)
        {
            if(P[i].AT < time && P[i].rem < P[smallest].rem && P[i].rem>0)
                smallest = i;
        }
        P[smallest].rem--;
        if (P[smallest].rem == 0)
        {
            count++;
            P[smallest].CT = time;
            P[smallest].TAT = P[smallest].CT - P[smallest].AT;
        }
        for(int i=0 ; i<n ; i++)
        {
            if(i!=smallest && P[i].AT < time && P[i].rem>0)
```

```
            {
                P[i].WT+=1;
            }
        }

    }
}

int main()
{
    process P[101];
    P[MAX].rem = 99999;
    int n,temp;
    printf("Number of processes : ");
    scanf("%d", &n);
    for(int i=0 ; i<n ; i++)
    {
        printf("Process %d:\n", i+1);
        printf("AT : ");
        scanf("%d", &temp);
        P[i].AT = temp;
        printf("BT : ");
        scanf("%d", &temp);
        P[i].BT = temp;
        P[i].id = i+1;
        P[i].CT = 0;
        P[i].WT = 0;
        P[i].TAT = 0;
        P[i].rem = P[i].BT;
    }
    quicksort(P, 0, n);
    simulate(P, n);
    printf("\n\n");
    printf("P\tAT\tBT\tCT\tTAT\tWT\n");
    float avgWT=0,avgTAT=0;
    for (int i = 0; i < n; ++i)
    {
        printf("%d\t%d\t%d\t%d\t%d\t%d\n", P[i].id,P[i].AT, P[i].BT, P[i].CT, P
[i].TAT, P[i].WT);
        avgTAT+=P[i].TAT;
        avgWT+=P[i].WT;
    }
    printf("Average Turn Around Time : %f\n", (avgTAT/n)*1.0);
    printf("Average Wating Time : %f\n", (avgWT/n)*1.0);
}

/*
OUTPUT

Number of processes : 4
Process 1:
AT : 1
BT : 8
Process 2:
AT : 2
BT : 4
Process 3:
AT : 3
BT : 9
Process 4:
AT : 4
BT : 5


P   AT  BT  CT  TAT WT
1   1   8   18  17  9
2   2   4   6   4   0
3   3   9   27  24  15
4   4   5   11  7   2
```

```
Average Turn Around Time : 13.000000
Average Wating Time : 6.500000
*/
```