```c
#include <stdio.h>
#include "queue.h"

int min(int a, int b)
{
    if(a > b)
        return b;
    return a;
}

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

void line(int n)
{
    for (int i = 0; i < n; i++)
        printf("=");
    printf("\n");
}

int main()
{
    int p, tq;
    printf("Enter no of processes: ");
    scanf("%d", &p);
    p = abs(p);

    printf("Enter Time Quantum : ");
    scanf("%d", &tq);
    tq = abs(tq);

    process arr[p];
    printf("Enter Arrival and Burst Time.\n");

    int total_time = 0;
    for (int i = 0; i < p; i++)
    {
        printf("Process %d : ", i + 1);
        scanf("%d", &arr[i].at);
        scanf("%d", &arr[i].bt);

        arr[i].at = abs(arr[i].at);
        arr[i].bt = abs(arr[i].bt);

        arr[i].pid = i;
        total_time += arr[i].bt;
    }

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

    int gantt[total_time];
    int queue_size = total_time / tq;
    init(queue_size);

    int clock = 0;
    int j = 0;

    int exe;
    int cq;
    bool flagExec = false;
    process executed;

    while (clock < total_time)
    {
        while (arr[j].at <= clock && j < p)
        {
            insert_last(arr[j]);
            j++;
        }

        if(flagExec)
        {
            insert_last(executed);
            flagExec = false;
        }

        process tmp = remove_first();
        if (tmp.pid == -1)
        {
            exe = -1;
            cq = 1;
        }
        else
        {
            exe = tmp.pid;

            cq = min(tq, tmp.bt);

            tmp.bt -= cq;
            if (tmp.bt > 0)
            {
                executed = tmp;
                flagExec = true;
            }
        }
        for(int i = clock; i < (clock + cq); i++)
        {
            gantt[i] = exe;
        }
        clock+=cq;
    }

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

    int ct[p], bt[p], tat[p], wt[p];
    for(int i=0;i<p;i++)
    {
        int total_bt = 0;
        int start = -1;
        int last;
        for(int j = 0;j<total_time;j++)
        {
            if(gantt[j] == arr[i].pid)
            {
                if(start == -1)
                {
                    start = j;
                }
                else
                {
                    last = j;
                }
                total_bt++;
            }
        }
        ct[i] = last +1;
        bt[i] = total_bt;
        tat[i] = ct[i] - arr[i].at;
        wt[i] = tat[i] - bt[i];
    }

    double avg_tat = 0, avg_wt = 0;
    line(66);
    printf("%10s|%10s|%10s|%10s|%10s|%10s|\n", "Process No", "A. T.","C. T.", "B. T.", "T. A. T.","W. T.");
    for(int i=0;i<p;i++)
    {
        printf("Process %2d|%10d|%10d|%10d|%10d|%10d|\n", arr[i].pid + 1, arr[i].at, ct[i], bt[i], tat[i], wt[i]);
        avg_tat += tat[i];
        avg_wt += wt[i];
    }
    line(66);
    avg_tat/=p;
    avg_wt/=p;
    printf("Average T. A. T. : %f\n", avg_tat);
    printf("Average W. T. : %f\n", avg_wt);

    return 0;
}

/*OUTPUT -
Enter no of processes: 5
Enter Time Quantum : 4
Enter Arrival and Burst Time.
Process 1 : 5 4
Process 2 : 0 9
Process 3 : 15 7
Process 4 : 12 12
Process 5 : 8 6
============================================================================
2 2 2 2 2 2 2 2 1 1 1 1 5 5 5 5 2 4 4 4 4 3 3 3 3 5 5 4 4 4 4 3 3 3 4 4 4 4 
============================================================================
==================================================================
Process No|     A. T.|     C. T.|     B. T.|  T. A. T.|     W. T.|
Process  2|         0|        17|         9|        17|         8|
Process  1|         5|        12|         4|         7|         3|
Process  5|         8|        27|         6|        19|        13|
Process  4|        12|        38|        12|        26|        14|
Process  3|        15|        34|         7|        19|        12|
==================================================================
Average T. A. T. : 17.600000
Average W. T. : 10.000000


Enter no of processes: 4
Enter Time Quantum : 20
Enter Arrival and Burst Time.
Process 1 : 0 53
Process 2 : 0 17
Process 3 : 0 68
Process 4 : 0 24
======================================================================================================================
======================================================================================================================
========================================================================================
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 
4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 
======================================================================================================================
======================================================================================================================
========================================================================================
==================================================================
Process No|     A. T.|     C. T.|     B. T.|  T. A. T.|     W. T.|
Process  1|         0|       134|        53|       134|        81|
Process  2|         0|        37|        17|        37|        20|
Process  3|         0|       162|        68|       162|        94|
Process  4|         0|       121|        24|       121|        97|
==================================================================
Average T. A. T. : 113.500000
Average W. T. : 73.000000
*/
```