## Program :-

```c
#include<stdio.h>

struct rrb {
    int pid, bt, at, ct, tt, wt;
};

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

                temp = p[j].bt;
                p[j].bt = p[j + 1].bt;
                p[j + 1].bt = temp;

                temp = p[j].at;
                p[j].at = p[j + 1].at;
                p[j + 1].at = temp;
            }
        }
    }
}

void roundrobin(struct rrb p[], int n) {
    int i, rbt[50], remain, flag = 0, sumt, timeq, k, temp;

    printf("\nEnter the TimeQuantum : ");
    scanf("%d", & timeq);

    for (i = 0; i < n; i++)
        rbt[i] = p[i].bt;
    remain = n;
    sumt = p[0].at;
    for (i = 0; remain != 0;) {
        if (rbt[i] <= timeq && rbt[i] > 0) {
            sumt = sumt + rbt[i];
            rbt[i] = 0;
            flag = 1;
        } else if (rbt[i] > 0) {
            rbt[i] = rbt[i] - timeq;
            sumt = sumt + timeq;
            for (k = i + 1; k < n; k++) {
                if (p[k].at <= sumt) {
                    temp = p[k - 1].pid;
                    p[k - 1].pid = p[k].pid;
                    p[k].pid = temp;

                    temp = p[k - 1].at;
                    p[k - 1].at = p[k].at;
                    p[k].at = temp;
```

```c
                temp = p[k - 1].bt;
                p[k - 1].bt = p[k].bt;
                p[k].bt = temp;

                temp = rbt[k - 1];
                rbt[k - 1] = rbt[k];
                rbt[k] = temp;

                flag = 2;
            } else
                break;
        }
    }
    if (rbt[i] == 0 && flag == 1) {
        remain--;
        p[i].ct = sumt;
        flag = 0;
    }

    if (i == n - 1)
        i = 0;
    else if (flag != 2 && p[i + 1].at <= sumt)
        i++;
    }
}

void tt(struct rrb p[], int n) {
    int i;
    for (i = 0; i < n; i++)
        p[i].tt = p[i].ct - p[i].at;
}

void wt(struct rrb p[], int n) {
    int i;
    for (i = 0; i < n; i++)
        p[i].wt = p[i].tt - p[i].bt;
}

void display(struct rrb p[], int n) {
    int i, j;
    struct rrb temp;
    printf("\npid\tat\tbt\tct\ttt\twt\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - 1 - i; j++) {
            if (p[j].pid > p[j + 1].pid) {
                temp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = temp;
            }
        }
    }
    for (i = 0; i < n; i++)
        printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].at, p[i].bt, p[i].ct, p[i].tt, p[i].wt);
}

void att(struct rrb p[], int n) {
```

```c
    int i, sum = 0;
    float att;
    for (i = 0; i < n; i++)
        sum += p[i].tt;
    att = (float) sum / (float) n;
    printf("\naverage turn around time = %.2f\n", att);
}

void awt(struct rrb p[], int n) {
    int i, sum = 0;
    float awt;
    for (i = 0; i < n; i++)
        sum += p[i].wt;
    awt = (float) sum / (float) n;
    printf("\naverage waiting time = %.2f\n\n", awt);
}

void main() {
    struct rrb p[50];
    int i, n;
    printf("\nenter how many processes : ");
    scanf("%d", & n);
    for (i = 0; i < n; i++) {
        printf("\nenter pid of #%d : ", i + 1);
        scanf("%d", & p[i].pid);
        printf("enter at    of #%d : ", i + 1);
        scanf("%d", & p[i].at);
        printf("enter bt    of #%d : ", i + 1);
        scanf("%d", & p[i].bt);
    }
    sort(p, n);
    roundrobin(p, n);
    tt(p, n);
    wt(p, n);
    display(p, n);
    att(p, n);
    awt(p, n);
}
```

Output :-

enter how many processes : 6
enter pid of #1 : 1
enter at    of #1 : 0
enter bt    of #1 : 5
enter pid of #2 : 2
enter at    of #2 : 1
enter bt    of #2 : 6
enter pid of #3 : 3
enter at    of #3 : 2
enter bt    of #3 : 3
enter pid of #4 : 4
enter at    of #4 : 3
enter bt    of #4 : 1
enter pid of #5 : 5
enter at    of #5 : 4
enter bt    of #5 : 5
enter pid of #6 : 6
enter at    of #6 : 6
enter bt    of #6 : 4
Enter the TimeQuantum : 4

| pid | at | bt | ct | tt | wt |
|-----|----|----|----|----|----|
| 1 | 0 | 5 | 17 | 17 | 12 |
| 2 | 1 | 6 | 23 | 22 | 16 |
| 3 | 2 | 3 | 11 | 9 | 6 |
| 4 | 3 | 1 | 12 | 9 | 8 |
| 5 | 4 | 5 | 24 | 20 | 15 |
| 6 | 6 | 4 | 21 | 15 | 11 |

average turn around time = 15.33

average waiting time = 11.33