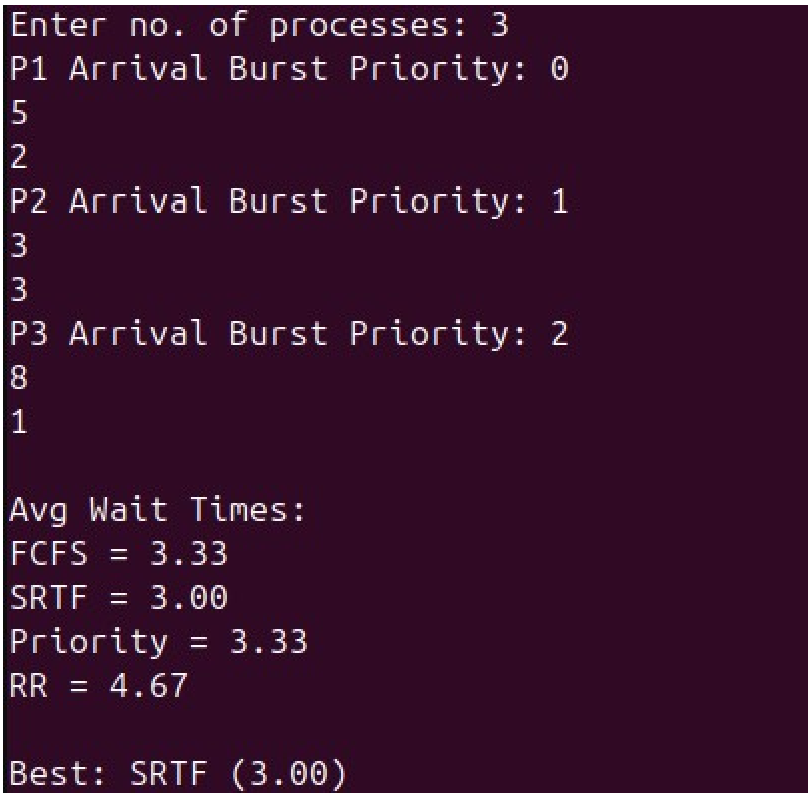
```
#include <stdio.h>
#include <limits.h>
#define Q 3
typedef struct {
int id, at, bt, pr, rt, ct, wt, tat;
} P;
void reset(P p[], P b[], int n) {
for (int i = 0; i < n; i++)
p[i] = b[i];
}
float fcfs(P p[], int n) {
int t = 0;
float avg = 0;
for (int i = 0; i < n; i++) {
if (t < p[i].at)
t = p[i].at;
t += p[i].bt;
p[i].ct = t;
p[i].wt = t - p[i].at - p[i].bt;
avg += p[i].wt;
}
return avg / n;
}
float srtf(P p[], int n) {
for (int i = 0; i < n; i++)
p[i].rt = p[i].bt;
int done = 0, t = 0, min, idx;
float avg = 0;
while (done < n) {
idx = -1;
min = INT_MAX;
for (int i = 0; i < n; i++) {
if (p[i].at <= t && p[i].rt > 0 && p[i].rt < min) {
min = p[i].rt;
idx = i;
}
}
if (idx == -1) {
t++;
continue;
}
p[idx].rt--;
t++;
if (p[idx].rt == 0) {
done++;
p[idx].ct = t;
p[idx].wt = t - p[idx].at - p[idx].bt;
avg += p[idx].wt;
}
}
return avg / n;
}
```

```
float npp(P p[], int n) {
int done = 0, t = 0, idx;
float avg = 0;
int f[n];
for (int i = 0; i < n; i++)
f[i] = 0;
while (done < n) {
int max = -1;
idx = -1;
for (int i = 0; i < n; i++) {
if (p[i].at <= t && !f[i] && p[i].pr > max) {
max = p[i].pr;
idx = i;
}
}
if (idx == -1) {
t++;
continue;
}
t += p[idx].bt;
p[idx].ct = t;
p[idx].wt = t - p[idx].at - p[idx].bt;
avg += p[idx].wt;
f[idx] = 1;
done++;
}
return avg / n;
}
float rr(P p[], int n) {
for (int i = 0; i < n; i++)
p[i].rt = p[i].bt;
int done = n, t = 0;
float avg = 0;
while (done) {
int flag = 0;

for (int i = 0; i < n; i++) {
if (p[i].at <= t && p[i].rt > 0) {
flag = 1;
if (p[i].rt > Q) {
t += Q;
p[i].rt -= Q;
} else {
t += p[i].rt;
p[i].rt = 0;
p[i].ct = t;
p[i].wt = t - p[i].at - p[i].bt;
avg += p[i].wt;
done--;
}
}
}
if (!flag)
t++;
}
return avg / n;
}
int main() {
int n;
printf("Enter no. of processes: ");
scanf("%d", &n);
P p[n], b[n];
for (int i = 0; i < n; i++) {
printf("P%d Arrival Burst Priority: ", i + 1);
scanf("%d %d %d", &p[i].at, &p[i].bt, &p[i].pr);
p[i].id = i + 1;
b[i] = p[i];
}

float f = fcfs(p, n); reset(p, b, n);
float s = srtf(p, n); reset(p, b, n);
float pr = npp(p, n); reset(p, b, n);
float r = rr(p, n);
printf("\nAvg Wait Times:\n");
printf("FCFS = %.2f\nSRTF = %.2f\nPriority = %.2f\nRR = %.2f\n", f, s, pr,r);
float min = f;
char algo[30] = "FCFS";
if (s < min) { min = s; sprintf(algo, "SRTF"); }
if (pr < min) { min = pr; sprintf(algo, "Priority"); }
if (r < min) { min = r; sprintf(algo, "RR"); }
printf("\nBest: %s (%.2f)\n", algo, min);
return 0;
}
```



```
Enter no. of processes: 3
P1 Arrival Burst Priority: 0
5
2
P2 Arrival Burst Priority: 1
3
3
P3 Arrival Burst Priority: 2
8
1

Avg Wait Times:
FCFS = 3.33
SRTF = 3.00
Priority = 3.33
RR = 4.67

Best: SRTF (3.00)
```