**EXPERIMENT: 07** ToConstruct a C program to implement a non-preemptive SJF algorithm.

**PROGRAM:**

#include <stdio.h>

struct Process {

int id, at, bt, ct, tat, wt, done;

};

int main() {

int n, t = 0, completed = 0;

printf("Enter number of processes: ");

scanf("%d", &n);

struct Process p[n];

for (int i = 0; i < n; i++) {

p[i].id = i + 1;

printf("P%d Arrival Time: ", i + 1);

scanf("%d", &p[i].at);

printf("P%d Burst Time: ", i + 1);

scanf("%d", &p[i].bt);

p[i].done = 0;

}

while (completed < n) {

int idx = -1, min\_bt = 9999;

for (int i = 0; i < n; i++) {

if (!p[i].done && p[i].at <= t && p[i].bt < min\_bt) {

min\_bt = p[i].bt;

idx = i;

}

}

if (idx != -1) {

t += p[idx].bt;

p[idx].ct = t;

p[idx].tat = p[idx].ct - p[idx].at;

p[idx].wt = p[idx].tat - p[idx].bt;

p[idx].done = 1;

completed++;

} else {

t++;

}

}

float avg\_wt = 0, avg\_tat = 0;

printf("\nP\tAT\tBT\tCT\tTAT\tWT\n");

for (int i = 0; i < n; i++) {

printf("P%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);

avg\_wt += p[i].wt;

avg\_tat += p[i].tat;

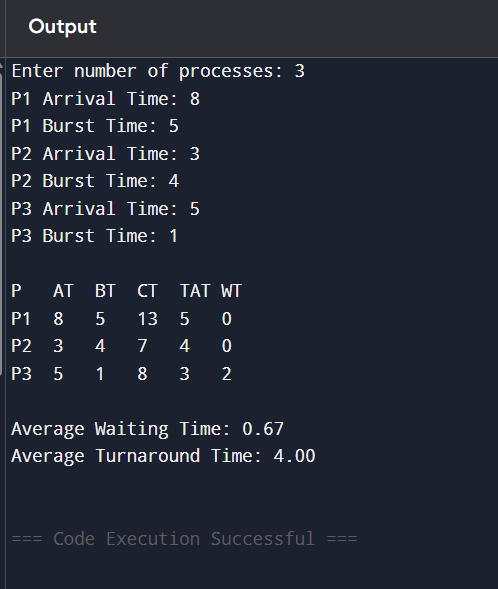
}

printf("\nAverage Waiting Time: %.2f", avg\_wt / n);

printf("\nAverage Turnaround Time: %.2f\n", avg\_tat / n);

return 0; }

**OUTPUT:**

****