#include <stdio.h>

#include <stdbool.h>

struct Process {

char pid[5];

int at, bt, ct, tat, wt;

bool completed;

};

int main() {

int n;

printf("Enter number of processes: ");

scanf("%d", &n);

struct Process p[n];

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

printf("Enter Process ID, Arrival Time & Burst Time: ");

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

p[i].completed = false;

}

int completed = 0, current\_time = 0;

printf("\nGantt Chart:\n");

while (completed < n) {

int idx = -1;

int min\_bt = 1e9;

// find process with shortest burst among arrived ones

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

if (!p[i].completed && p[i].at <= current\_time) {

if (p[i].bt < min\_bt) {

min\_bt = p[i].bt;

idx = i;

}

}

}

if (idx == -1) {

// CPU idle, jump to next arrival

int next\_arrival = 1e9;

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

if (!p[i].completed && p[i].at < next\_arrival) {

next\_arrival = p[i].at;

}

}

printf("| %d IDLE %d ", current\_time, next\_arrival);

current\_time = next\_arrival;

} else {

// schedule selected process

int start = current\_time;

int finish = current\_time + p[idx].bt;

p[idx].ct = finish;

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

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

p[idx].completed = true;

printf("| %d %s %d ", start, p[idx].pid, finish);

current\_time = finish;

completed++;

}

}

printf("|\n");

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

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

printf("%s\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].tat, p[i].wt);

}

return 0;

}

#include <stdio.h>

struct P{int at,bt,rt,ct,tat,wt,id;}p[10];

int main(){

int n,t=0,done=0; printf("Enter no. of processes: "); scanf("%d",&n);

for(int i=0;i<n;i++){p[i].id=i+1;printf("AT BT for P%d: ",i+1);

scanf("%d%d",&p[i].at,&p[i].bt);p[i].rt=p[i].bt;}

printf("\nGantt:\n");

while(done<n){

int idx=-1,min=1e9;

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){printf("|%d IDLE ",t);t++;}

else{printf("|%d P%d ",t,p[idx].id);p[idx].rt--;t++;

if(!p[idx].rt){p[idx].ct=t;p[idx].tat=t-p[idx].at;p[idx].wt=p[idx].tat-p[idx].bt;done++;}}}

printf("|%d|\n\nPID AT BT CT TAT WT\n",t);

float aT=0,aW=0;

for(int i=0;i<n;i++){printf("P%d %d %d %d %d %d\n",p[i].id,p[i].at,p[i].bt,p[i].ct,p[i].tat,p[i].wt);

aT+=p[i].tat;aW+=p[i].wt;}

printf("Avg TAT=%.2f Avg WT=%.2f\n",aT/n,aW/n);

}