**PROGRAM CODE**

#include<stdio.h>  
#include<stdlib.h>  
  
#define STATE\_UNKNOWN 0  
#define STATE\_READY 1  
#define STATE\_RETURNED 2  
  
struct entry

{  
 int AT, BT, CT, TAT, WT, ST, priority, state;  
 char Name[20];  
} pChart[10];  
int n, readyQue[10], ready\_top = 0, arrSort[10];  
  
void swap (int\* list, int i1, int i2)

{  
 int temp = list[i1];  
 list[i1] = list[i2];  
 list[i2] = temp;  
}  
  
void enque(int id)

{  
 if (pChart[id].state != STATE\_UNKNOWN)

return;  
 pChart[id].state = STATE\_READY;  
 readyQue[ready\_top] = id;   
 for(int j = ready\_top++; j > 0 && pChart[readyQue[j-1]].priority <= pChart[readyQue[j]].priority; j--)  
 swap(readyQue, j-1, j);  
}  
  
int nextProcessId()

{  
 if (ready\_top == 0) return -1;  
 return readyQue[--ready\_top];  
}  
  
int main ()

{  
 printf("Number of Processes >> ");  
 scanf("%d", &n);  
  
 for (int i = 0; i < n; i++)

{  
 printf("Process %d (PID\_PRIORITY\_AT\_BT) >> ", i+1);  
 scanf("%s%d%d%d", pChart[i].Name, &pChart[i].priority, &pChart[i].AT, &pChart[i].BT);  
 arrSort[i] = i;  
 pChart[i].state = STATE\_UNKNOWN;  
 pChart[i].Name[7] = '\0';  
 }  
   
 for (int i = 1; i < n; i++)  
 for (int j = i; j > 0 && pChart[arrSort[j-1]].AT > pChart[arrSort[j]].AT; j--)  
 swap(arrSort, j-1, j);  
  
 int pStarted = 0, gEntry[20], gTop = 0, t\_TAT = 0, t\_WT = 0;  
 for (int cTime = 0; pStarted < n; )

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

{  
 if (pChart[arrSort[i]].state != STATE\_UNKNOWN) continue;  
 if (pChart[arrSort[i]].AT > cTime) break;  
 enque(arrSort[i]);  
 }  
  
 int pid = nextProcessId();  
 struct entry \*cp = &pChart[pid];  
 if (pid > -1)

{  
 cp->ST = cTime;  
 cTime += cp->BT;  
 cp->CT = cTime;  
 cp->TAT = cp->CT - cp->AT;  
 cp->WT = cp->TAT - cp->BT;  
 t\_TAT += cp->TAT;  
 t\_WT += cp->WT;  
 gEntry[gTop++] = pid;  
 cp->state = STATE\_RETURNED;  
 pStarted++;  
 }

else

{  
 if (gEntry[gTop-1] != -1)

gEntry[gTop++] = -1;  
 cTime++;  
 }  
 }  
   
 printf("\n+---------+----------+----+----+----+-----+----+\n");  
 printf("| PROCESS | PRIORITY | AT | BT | CT | TAT | WT |\n");  
 printf("+---------+----------+----+----+----+-----+----+\n");  
 for (int i = 0; i < n; i++)

{  
 printf("|%9s|%10d|%4d|%4d|", pChart[i].Name, pChart[i].priority, pChart[i].AT, pChart[i].BT);  
 printf("%4d|%5d|%4d|\n", pChart[i].CT, pChart[i].TAT, pChart[i].WT);  
 }  
 printf("+---------+----------+----+----+----+-----+----+\n");  
   
 printf("\nAvg TAT = %f\nAvg WT = %f\n", (float)t\_TAT/n, (float)t\_WT/n);  
}  
**OUTPUT**

