```
#include <stdio.h>
#include <string.h>
int Q[100],front=-1,rear=-1;
struct process
{  char name[20];
   int at,tt,bt,wt,status,left,ct;}p[20],temp;

struct done
{  char name[20];
   int st,ct;}d[20];

void ENQUEUE(int j)
{  if(front==-1 && rear==-1)
   {  front++;   }
   rear++;
   Q[rear] = j;   }

int DEQUEUE()
{  int item;
   item = Q[front];
   if(front == rear)
   {  front = -1;
      rear = -1;    }
   else
   {  front++;   }
   return(item);  }

void main()
{  int n,i,j,idle=0,k,num,ls,t;
   float avwt=0,avtt=0;
   printf("ENTER THE NUMBER OF PROCESSES : ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {  printf("\nENTER DETAILS OF PROCESS %d",i+1);
      printf("\nPROCESS NAME : ");
      scanf(" %s",p[i].name);
      printf("ARRIVAL TIME : ");
      scanf("%d",&p[i].at);
      printf("BURST TIME : ");
      scanf("%d",&p[i].bt);
      p[i].left = p[i].bt;
      p[i].status = 0;   }
   printf("\nENTER THE TIME QUANTUM : ");
   scanf("%d",&t);

   for(i=0,num=0,ls=0;ls<n;)
   {  for(j=0;j<n;j++)
      {  if(p[j].status==0 && p[j].at<=i)
         {  ENQUEUE(j);
            p[j].status = 1;   }     }
      if(idle==0 && front == -1)
      {  strcpy(d[num].name,"Idle");
         d[num].st = i;
         idle = 1;
         i++;   }
      else if(front!=-1)
      {  if(idle==1)
         {  d[num].ct = i;
```

```
        idle = 0;
        num++;    }
      k = DEQUEUE();
      d[num].st = i;
      strcpy(d[num].name,p[k].name);
      if(p[k].left<=t)
      {  d[num].ct = i+p[k].left;
        p[k].ct = d[num].ct;
        i = d[num].ct;
        p[k].tt = i - p[k].at;
        p[k].wt = p[k].tt - p[k].bt;
        p[k].status = 2;
        ls++;
        num++;    }
    else if(p[k].left>t)
    {  d[num].ct = i+t;
      i = d[num].ct;
      p[k].left = p[k].left-t;
      num++;
      for(j=0;j<n;j++)
      {  if(p[j].status==0 && p[j].at<=i)
        {  ENQUEUE(j);
          p[j].status = 1;  }    }
      ENQUEUE(k);        }       }
    else
    {  i++;     }    }

  printf("\nPROCESS NAME\tCOMPLETION TIME (ms)\tWAITING TIME (ms)\tTURNAROUND
TIME (ms)\n\n");
  for(i=0;i<n;i++)
  {  printf("   %s\t\t\t%d\t\t\t%d\t\t\t%d\n",p[i].name,p[i].ct,p[i].wt,p[i].tt);
    avwt+=p[i].wt;
    avtt+=p[i].tt;    }

  printf("\n\nGANTT CHART ");
  printf("\n\t--------------------------------------------------------------------------\n\t");
  for(i=0;i<num;i++)
  {  printf("|");
    printf("P%s\t",d[i].name);    }
  printf(" |");
  printf("\n\t--------------------------------------------------------------------------\n\t");
  for(i=0;i<num;i++)
  {  printf("%d\t",d[i].st);   }
  printf("%d\t",d[num-1].ct);
  printf("\n\nAVERAGE WAITING TIME : %f",(avwt/n));
  printf("\nAVERAGE TURNAROUND TIME : %f\n",(avtt/n));     }
```