**Code:**

#include<stdio.h>

int main() {

int count=0;

int time\_quantum,limit,total\_time=0;

float avg\_turnaround\_time,avg\_waiting\_time;

printf("Enter number of processess\n");

scanf("%d",&limit);

int process[limit],arrival\_time[limit],burst\_time[limit],bursttime[limit],queue\_number[limit],turnaround\_time[limit],waiting\_time[limit];

printf("Enter time quantum in multiples of two\n");

scanf("%d",&time\_quantum);

if((time\_quantum%2)!=0) {

printf("Time quantum entered is not in multiples of two\n");

return 0;

}

for(int j=0;j<limit;j++) {

printf("Enter process %d process id\n",j+1);

scanf("%d",&process[j]);

printf("Enter process %d arrival time\n",j+1);

scanf("%d",&arrival\_time[j]);

printf("Enter process %d burst time\n",j+1);

scanf("%d",&burst\_time[j]);

printf("Enter process %d priority\n",j+1);

scanf("%d",&queue\_number[j]);

total\_time=total\_time+burst\_time[j];

bursttime[j]=burst\_time[j];

}

printf("\n\n\n\n");

while(count<total\_time){

for(int j=0;j<limit;j++) {

if( (arrival\_time[j]==0 && burst\_time[j]>0 && queue\_number[j]==0) || (arrival\_time[j]<=count && burst\_time[j]>0 && queue\_number[j]==0) ) {

if(burst\_time[j]!=1) {

burst\_time[j]=burst\_time[j]-time\_quantum;

count=count+time\_quantum;

if(burst\_time[j]==0 || burst\_time[j]<0) {

turnaround\_time[j]=count-arrival\_time[j];

waiting\_time[j]=turnaround\_time[j]-bursttime[j];

printf("Process with process id %d has finished with turnaround time %d and waiting time %d\n\n",process[j],turnaround\_time[j],waiting\_time[j]);

}

}

else if(burst\_time[j]==1) {

burst\_time[j]=burst\_time[j]+(time\_quantum-(time\_quantum+1));

count=count+(time\_quantum-(time\_quantum-1));

if(burst\_time[j]==0) {

turnaround\_time[j]=count-arrival\_time[j];

waiting\_time[j]=turnaround\_time[j]-bursttime[j];

printf("Process with process id %d has finished with turnaround time %d and waiting time %d\n\n",process[j],turnaround\_time[j],waiting\_time[j]);

}

}

}

}

for(int j=0;j<limit;j++) {

if(arrival\_time[j]<=count && queue\_number[j]==1 && burst\_time[j]>0) {

int x=1;

while(x==1 && burst\_time[j]>0 ) {

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

if(arrival\_time[i]<=count && burst\_time[i]>0 && queue\_number[i]==0) {

x=0;

}

}

if(x==1) {

if(burst\_time[j]!=1) {

burst\_time[j]=burst\_time[j]-time\_quantum;

count=count+time\_quantum;

if(burst\_time[j]==0 || burst\_time[j]<0) {

turnaround\_time[j]=count-arrival\_time[j];

waiting\_time[j]=turnaround\_time[j]-bursttime[j];

printf("Process with process id %d has finished with turnaround time %d and waiting time %d\n\n",process[j],turnaround\_time[j],waiting\_time[j]);

}

}

else if(burst\_time[j]==1) {

burst\_time[j]=burst\_time[j]+(time\_quantum-(time\_quantum+1));

count=count+(time\_quantum-(time\_quantum-1));

if(burst\_time[j]==0) {

turnaround\_time[j]=count-arrival\_time[j];

waiting\_time[j]=turnaround\_time[j]-bursttime[j];

printf("Process with process id %d has finished with turnaround time %d and waiting time %d\n\n",process[j],turnaround\_time[j],waiting\_time[j]);

}

}

}

}

}

}

}

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

avg\_turnaround\_time=avg\_turnaround\_time+turnaround\_time[i];

avg\_waiting\_time=avg\_waiting\_time+waiting\_time[i];

}

printf("Average turnaround time is %f\n\nAverage waiting time is %f\n\n",(avg\_turnaround\_time/limit),(avg\_waiting\_time/limit));}