9) Write a program to implement non-preemptive priority based scheduling algorithm.

```c
#include<stdio.h>
int main()
{
   int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp;
   float avg_wt,avg_tat;
   printf("Enter Total Number of Process:");
   scanf("%d",&n);
   printf("\nEnter Burst Time and Priority\n");
  for(i=0;i<n;i++)
  {
  printf("\nP[%d]\n",i+1);
  printf("Burst Time:");
  scanf("%d",&bt[i]);
  printf("Priority:");
  scanf("%d",&pr[i]);
  p[i]=i+1; //contains process number
  }
//sorting burst time, priority and process number in ascending order using selection sort
for(i=0;i<n;i++)
{
  pos=i;
  for(j=i+1;j<n;j++)
{
  if(pr[j]<pr[pos])
  pos=j;
}
temp=pr[i];
pr[i]=pr[pos];
pr[pos]=temp;
temp=bt[i];
bt[i]=bt[pos];
bt[pos]=temp;
temp=p[i];
p[i]=p[pos];
p[pos]=temp;
}
wt[0]=0; //waiting time for first process is zero
//calculate waiting time
for(i=1;i<n;i++)
{
```

```c
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
total+=wt[i];

}
avg_wt=(float)total/n; //average waiting time
total=0;
printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
   tat[i]=bt[i]+wt[i]; //calculate turnaround time
   total+=tat[i];
   printf("\nP[%d]\t\t %d\t\t\t %d\t\t\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
}
avg_tat=(float)total/n; //average turnaround time
printf("\n\nAverage Waiting Time=%f",avg_wt);
printf("\nAverage Turnaround Time=%f\n",avg_tat);
return 0;
}
```

OUTPUT :

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> g++ -o main priority.cpp &&./main
Enter Total Number of Process:4

Enter Burst Time and Priority

P[1]
Burst Time:4
Priority:2

P[2]
Burst Time:5
Priority:1

P[3]
Burst Time:7
Priority:3

P[4]
Burst Time:6
Priority:4

Process   Burst Time      Waiting Time      Turnaround Time
P[2]          5               0                    5
P[1]          4               5                    9
P[3]          7               9                    16
P[4]          6               16                   22

Average Waiting Time=7.500000
Average Turnaround Time=13.000000
> []
```