Q- Write a program in C to impliment FCFS

```
# include <stdio.h>
int main()
{
int n, bt[20], wt[20], tat[20], avwt =0, avtat = 0,i, j:
·Printf("Enter total number numbes of processes (maximum20):")
Scanf ("%d",&n);
printf("\n Enter process Burst Time \n");
for (i=0; i<n; i++);
{
printf("P[%d]:", i+1);
Scanf ("%d",&bt[i]);
}
wt[0]=0;
for(i=1; i<n; i++)
{
wt[i]=0;
for(j=0; j<i; j++)
        wt[i]+ =bt[j];
}
printf("\n Process\t\tBurst Time/twaiting Time\t Turn
for (i=0; i<n; i++) ·                          round Time");
{
tat[i] = bt[i]+ wt [i];
avwt + = wt [i];
av tat + = tat [i];
printf(" \nP[%d]\t\t%d\t\t%d\t\t%d", i+1,
}                                    wt[i], tat[i]);
avwt/=i;
avtat/='i;
```

```
printf("\n\n Average age waiting time %.",avw
printf("\nAverage Turnaround Time: %d", avtat);
return 0;
}
```

Output

```
Processes Burst time waiting time Turn around time
   Process        Burst time     waiting time    Turnaround time
   P1               24              0               24
   P2               3               24              27
   P3               3               27              30
Average Waiting Time: 17
Average Turnaround time: 127
```

Q- Write a program in c to impliment [illegible]SFS.

```
#include <stdio.h>
 int main()
 {
 int bt[20],P[20],wt[20],tat[20], i, j, n, total = 0, Pos, temps
 float avg-wt, avg, tat;
 printf ("Enter number of process:");
  scanf ("%d", &n);
  printf ("n Enter Burst Time: n");
 for (i= 0; i<n; i++)
  {
  printf ("P%d: ", i+1);
 Scanf ("%d", &bt [i]);
 P[i] = i+1;
 P[i] = i+1;
  }
  for (i=0; i<n; i++)
  {
   Pos = i;
  for (j = i+1; j<n; j++)
   {
    if (bt [j] < bt [pos])
    Pos = j;
   }
 temp = bt [i];
  bt[i] = bt [pos];
 bt [pos] = temp;
  temp = P[i];
bt[i] = bt[pos];
bt[pos] = temp;
temp = p[i];
p[i] = p[pos];
p[pos] = temp;
}
wt[0] = 0;
for(i=1; i<n; i++)
 {
  wt[i] = 0;
 for(j=0; j<i; j++)
  wt[i] += bt[j];
  total += wt[i];
 }
 avg-wt = (float) total / n;
 total = 0;
 Printf("\n Process Burst Time \twaiting Turnaround Time");
for(i=0; i<n; i++)
{
tat[i] = bt[i] + wt[i];
total += tat[i];
Printf("\np%d\t %d\t %d\t\t %d", p[i], bt[i], wt[i], tat[i])
}
avg-tat = (float) total / n;
Printf("\n\n Average Waiting time = %f", avg-wt);
printf("\n Average turnaround Time = %f\n", avg-tat);
}
```

```
{
 temp[i] = temp [i] - quant;
 sum = sum + quant ;
}
 if (temp[i] = = 0 && count = =1)
  {
   y - - ;
   printf ("\n Process No[%d] \t \t %d\t\t\t\t %d\t\t\t
           %d 0", i+1, bt[i], sum-at [i], sum - at [i] - bt [i]);
            wt = wt + sum -at [i] - bt [i];
            tat = tat + sum - at [i];
            count =0;
     }
      if (i= = NOP -1)
     {
         i=0;
    }
    else if (at (i+1) <=sum )
    {
       i++;
    }
    else
     {
       i=0;
     }
  }
}
```

Output

```
Total number of processes in the system :2
Enter the Arrival and Burst time of the Process [1]
Enter Arrival time : 0
Enter Burst time : 5
Enter the arrival and Burst time the of the Process[2]
Enter Arrival time : 1
Enter Burst time : 4
```

Output

```
Process Burst   time Waiting   time Turn around time

Process         Bt             Wt             TT
P4              1              0              1
P5              2              1              3
P2              3              3              6
P1              4              3              10
P3              2              6              17
                               10
Average Waiting time = 4.00000
Averag Turnoround time = 7.400000
```

Output

```
Total number of processes in the system :2
Enter the Arrival and Burst time of the Process [1]
Enter Arrival time : 0
Enter Burst time : 5

Enter the arrival and Burst time the of the Process[2]
Enter Arival time : 1
Enter Burst time : 4
```

Q Write a program in C to impliment P.S.

```
#include<stdio.h>
void main()
{
 int x,n,p[10],pp[10]w[10],t[10],awt,atat,i;
 Printf("Enter the number of process:");
 Scanf("%d",&n);
 printf("\n Enter process: time priorities\n");
 for(i=0; i<n; i++)
 {
  printf("\n Process no %d :", i+1);
  Scanf("%d", & pt[i], & pp[i]);
  P[i] = i+1;
  }
  for(i=0; i<n-1 ; i++)
  {
   for(int j=i+1; j<n; j++)
   { if(pp[i] < pp[j])
     {
      x = pp[i];
      pp[i] = pp[j];
       pp[j] = x;
        x = pt[i];
         pt[i] = pt[j];
          pt[j] = x;
           x = p[i];
            p[i] = p[j];
             p[j] = 0;
       }
    }
}
 w[0] = 0;
awt = 0;
t[0] = pt[0];
atat = t[0];
for(i=1; i<n; i++)
 {
  w[i] = t[i-1];
  awt += w[i];
  t[i] = w[i] + pt[i];
  atat += t[i];
 }
Printf("\n\n job \t Burst Time \t wait Time \t Turn around time Priority\n
                                                                    ");
for (i=0; i<n; i++)
   printf("\n %d \t\t %d \t\t %d \t\t %d \n", p[i], pt[i], w[i], t[i]
                                        pp[i]);
   awt /= n;
   atat /= n;
 Printf("\n Average Wait Time : %d \n", awt);
 Printf("\n Average Turn around Time : %d \n", atat);
  getch()
 }
```

Output

```
    Enter the number of process: 4
    Enter process : time priorities
Job   Bt       wt      TAT      P
 4     6        0       6       4
 3     5        6       11      3
 2     4        11      15      2
 1     3        13      18      1
Average wait time: 8
Average Turn Around time: 12
```

Q- Write a program in C to impliment RR Scheduling

```
# include <stdio .h>
void main()
{
  int i, NOP, sum = 0, count=0, y, quant, wt = 0, tat = 0, at[10], bt[10], temp
                                                              [10];
      float avg-wt, avg-tat;
      printf("Total number of process in the system:");
      scanf("%d", & NOP);
        y = NOP;
      for(i = 0; i<NOP; i++)
      {
        Printf("\n Enter the Arrival and Burst time of the Process [%d]\n", i+1);
        Printf("Enter arrival time:\t");
        sanf("%d", & at[i]);
        printf("\n Enter Burst time: \t");
        scanf("%d", & bt[i]);
         temp[i] = bt [i];
      }
      Printf("Enter the time Quantum for the process: \t");
      Scanf("%d" & quant);
      Printf("\n Process No\t\t Burst Time \t\t TAT \t\t waiting
               Time");
      for (sum= 0, i = 0; y!=0;)
      {
        if (temp[i] <= quant && temp [i] > 0
      {
        sum = sum + temp [i];
        temp [i] = 0;
        count = 1;
        }
        else if (temp[i] > 0)
```