PROGRAM FOR FCFS

```
#include<stdio.h>
#include<stdlib.h>
struct processor
{
  float bt;
  float at;
  float cmp;
  float tat;
  float wt;
};
void main()
{
  int n;
  float sum;
  printf("\nenter the number of processors\n");
  scanf("%d",&n);
  struct processor p[n];
  printf("\nenter the burst time for %d processors \n",n);
  for(int i=0;i<n;i++)
  {
     scanf("%f",&p[i].bt);
  }
  printf("\nenter the arrival time for %d processors \n",n);
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p[i].at);
  }

printf("\n*****************************************************\n")
;
  printf("\nprocessor\tarrival time\t\tburst time\n");
  for(int i=0;i<n;i++)
  {
    printf("%d\t\t%f\t\t%f\t\n",i+1,p[i].at,p[i].bt);
  }

printf("\n*****************************************************\n")
;
  float temp;
  for(int i=0;i<n;i++)
  {
    for(int j=i+1;j<n;j++)
    {
      if(p[i].at>p[j].at)
      {
        temp=p[i].bt;
        p[i].bt=p[j].bt;
        p[j].bt=temp;

        temp=p[i].at;
        p[i].at=p[j].at;
        p[j].at=temp;
      }
    }
  }
  p[0].cmp=p[0].bt;
  for(int i=1;i<n;i++)
  {
    p[i].cmp=p[i].bt+p[i-1].cmp;
  }
  for(int i=0;i<n;i++)
  {
    p[i].tat=p[i].cmp-p[i].at;
  }
  sum=0;
  for(int i=0;i<n;i++)
  {
    sum=sum+p[i].tat;
  }
  float avg_tat=sum/n;

  printf("\naverage turn around time=%f",avg_tat);
  for(int i=0;i<n;i++)
  {
    p[i].wt=p[i].tat-p[i].bt;
  }
  sum=0;
  for(int i=0;i<n;i++)
  {
    sum=sum+p[i].wt;
  }
  float avg_wt=sum/n;
  printf("\naverage waiting time=%f\n",avg_wt);
}
```

OUTPUT

```
sahal@kali:~/bash_script$ ./a.out

enter the number of processors
5

enter the burst time for 5 processors
2
```

```
6
4
9
12

enter the arrival time for 5 processors
0
1
2
3
4

***************************************************

processor       arrival time            burst time
1               0.000000                2.000000
2               1.000000                6.000000
3               2.000000                4.000000
4               3.000000                9.000000
5               4.000000                12.000000

***************************************************

average turn around time=13.200000
average waiting time=6.600000
```

PROGRAM CODE FOR SJF

```
#include<stdio.h>
#include<stdlib.h>
struct processor
{
  float bt;
  float at;
  float cmp;
  float tat;
  float wt;
};
void main()
{
  int n;
  float sum;
  printf("\nenter the number of processors\n");
  scanf("%d",&n);
  struct processor p1[n];
  printf("\nenter the burst time for %d processors \n",n);
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p1[i].bt);
  }
  printf("\nenter the arrival time\n");
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p1[i].at);
  }

printf("\n**************************************************\n")
;
  printf("\nprocessor\tarrival time\t\tburst time\n");
  for(int i=0;i<n;i++)
  {
    printf("%d\t\t%f\t\t%f\t\n",i+1,p1[i].at,p1[i].bt);
  }

printf("\n**************************************************\n")
;
  float temp;
  for(int i=0;i<n;i++)
  {
    for(int j=i+1;j<n;j++)
    {
      if(p1[i].at>p1[j].at)
      {
        temp=p1[i].bt;
        p1[i].bt=p1[j].bt;
        p1[j].bt=temp;

        temp=p1[i].at;
        p1[i].at=p1[j].at;
        p1[j].at=temp;
      }
    }
  }
  float tot_cmp=0;
  int k=1;
  for(int i=0;i<n;i++)
  {
    float tot_cmp=tot_cmp+p1[i].bt;
    for(int j=k;j<n;j++)
    {
      float min=p1[k].bt;
      if(p1[j].at<tot_cmp&&p1[j].bt<min)
      {
        temp=p1[k].bt;
        p1[k].bt=p1[j].bt;
        p1[j].bt=temp;

        temp=p1[k].at;
        p1[k].at=p1[j].at;
        p1[j].at=temp;
      }
    }
    k++;
  }
  p1[0].cmp=p1[0].bt;
  for(int i=0;i<n;i++)
  {
    p1[i].cmp=p1[i-1].cmp+p1[i].bt;
  }
  for(int i=0;i<n;i++)
  {
    p1[i].tat=p1[i].cmp-p1[i].at;
  }
  for(int i=0;i<n;i++)
  {
    p1[i].wt=p1[i].tat-p1[i].bt;
  }
  sum=0;
  for(int i=0;i<n;i++)
  {
    sum+=p1[i].tat;
  }
  float avg_tat=sum/n;
  printf("\naverage turn around time=%f\n",avg_tat);
  sum=0;
  for(int i=0;i<n;i++)
  {
    sum+=p1[i].wt;
  }
  float avg_wt=sum/n;
  printf("\naverage waiting time=%f\n",avg_wt);
}
```

OUTPUT FOR SJF

```
sahal@kali:~/bash_script$ ./a.out

enter the number of processors
5

enter the burst time for 5 processors
6
```

```
2
8
3
4

enter the arrival time
2
5
1
0
4

***************************************************

processor       arrival time            burst time
1               2.000000                6.000000
2               5.000000                2.000000
3               1.000000                8.000000
4               0.000000                3.000000
5               4.000000                4.000000

***************************************************

average turn around time=9.800000

average waiting time=5.200000
```

PROGRAM CODE PRIORITY

```
#include<stdio.h>
#include<stdlib.h>
struct processor
{
  float bt;
  float at;
  float cmp;
  float tat;
  float wt;
  float pr;
};
void main()
{
  int n;
  float sum;
  printf("\nenter the number of processors\n");
  scanf("%d",&n);
  struct processor p1[n];
  struct processor p;
  printf("\nenter the burst time for %d processors \n",n);
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p1[i].bt);
  }
  printf("\nenter the arrival time\n");
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p1[i].at);
  }
  printf("\nenter the priority for %d processors\n",n);
  for(int i=0;i<n;i++)
  {
    scanf("%f",&p1[i].pr);
  }

printf("\n**************************************************
******\n");
  printf("\nprocessor\tarrival time\t\tburst time\t\tpriority\n");
  for(int i=0;i<n;i++)
  {

printf("%d\t\t%f\t\t%f\t\t%f\n",i+1,p1[i].at,p1[i].bt,p1[i].pr);
  }

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

  float temp;
  for(int i=0;i<n;i++)
  {
    for(int j=i+1;j<n;j++)
    {
        if(p1[i].at>p1[j].at)
         {
          temp=p1[i].bt;
          p1[i].bt=p1[j].bt;
          p1[j].bt=temp;

          temp=p1[i].at;
          p1[i].at=p1[j].at;
          p1[j].at=temp;

          temp=p1[i].pr;
          p1[i].pr=p1[j].pr;
          p1[j].pr=temp;
         }
```

```
   }
}
float tot_cmp=0;
int k=1;
float min;
for(int i=0;i<n;i++)
{
  tot_cmp=tot_cmp+p1[i].bt;
  for(int j=k;j<n;j++)
  {
    min=p1[k].pr;
    if(p1[j].at<tot_cmp&&p1[j].pr<min)
    {
      temp=p1[k].bt;
      p1[k].bt=p1[j].bt;
      p1[j].bt=temp;

      temp=p1[k].at;
      p1[k].at=p1[j].at;
      p1[j].at=temp;

      temp=p1[k].pr;
      p1[k].pr=p1[j].pr;
      p1[j].pr=temp;
    }
  }
  k++;
}
p1[0].cmp=p1[0].bt;

for(int i=1;i<n;i++)
{
  p1[i].cmp=p1[i-1].cmp+p1[i].bt;
}
for(int i=0;i<n;i++)
{
  p1[i].tat=p1[i].cmp-p1[i].at;
}
for(int i=0;i<n;i++)
{
  p1[i].wt=p1[i].tat-p1[i].bt;
}
sum=0;
for(int i=0;i<n;i++)
{
  sum+=p1[i].tat;
}
float avg_tat=sum/n;
```

```
  printf("\naverage turn around time=%f\n",avg_tat);
  sum=0;
  for(int i=0;i<n;i++)
  {
    sum+=p1[i].wt;
  }
  float avg_wt=sum/n;
  printf("\naverage waiting time=%f\n",avg_wt);
}
```

OUTPUT FOR PRIORITY

```
sahal@kali:~/bash_script$ ./a.out

enter the number of processors
5

enter the burst time for 5 processors
4
3
7
4
2

enter the arrival time
0
0
6
11
12

enter the priority for 5 processors
1
2
1
3
2

********************************************************

processor       arrival time            burst time
priority
1               0.000000                4.000000
1.000000
2               0.000000                3.000000
2.000000
3               6.000000                7.000000
```

```
1.000000
4               11.000000               4.000000
3.000000
5               12.000000               2.000000
2.000000

*******************************************************

average turn around time=6.400000

average waiting time=2.400000
```

PROGRAM CODE FOR ROUND ROBIN

```
#include<stdio.h>
#include<stdlib.h>
struct processor
{
  float bt;
  float at;
  float cmp;
  float tat;
  float wt;
};
void main()
{
  int n;
  float sum;
  printf("\nenter the number of processors\n");
  scanf("%d",&n);
  struct processor p1[n];
  float burst[n];
  printf("\nenter the burst time for %d processors \n",n);
  for(int i=0;i<n;i++)
  {
     scanf("%f",&p1[i].bt);
     burst[i]=p1[i].bt;
  }
  printf("\nenter the arrival time for %d processors\n",n);
  for(int i=0;i<n;i++)
  {
     scanf("%f",&p1[i].at);
  }

printf("\n*****************************************************\n")
;
  printf("\nprocessor\tarrival time\t\tburst time\n");
  for(int i=0;i<n;i++)
  {
    printf("%d\t\t%f\t\t%f\t\n",i+1,p1[i].at,p1[i].bt);
  }

printf("\n****************************************************\n")
;
  int tq;
  printf("\nenter the time quantum\n");
  scanf("%d",&tq);
  float temp;
  for(int i=0;i<n;i++)
  {
    for(int j=i+1;j<n;j++)
    {
      if(p1[i].at>p1[j].at)
      {
        temp=p1[i].bt;
        p1[i].bt=p1[j].bt;
        p1[j].bt=temp;

        temp=p1[i].at;
        p1[i].at=p1[j].at;
        p1[j].at=temp;
      }
    }
  }

  float tot_bt=0;
  int i=0;
  int count;
  int x=n;
  float waiting_time=0;
  float turn_around_time=0;

  while(x!=0)
  {
    if(burst[i]<=tq&&burst[i]>0)
    {
      tot_bt=tot_bt+burst[i];
      burst[i]=0;
      count=1;
    }
    else if(burst[i]>0)
    {
      burst[i]=burst[i]-tq;
      tot_bt=tot_bt+tq;
    }
    if(burst[i]==0&&count==1)
    {
      x--;
      waiting_time=waiting_time+tot_bt-p1[i].at-p1[i].bt;
      turn_around_time=turn_around_time+tot_bt-p1[i].at;
      count=0;
    }
    if(i==n-1)
    {
      i = 0;
    }
    else if(p1[i+1].at<=tot_bt)
    {
      i++;
    }
    else
    {
      i = 0;
    }
  }
  float avg_tat=turn_around_time/n;
  float avg_wt=waiting_time/n;
  printf("\nturn around timie=%f\n",avg_tat);
  printf("\nwaiting time=%f\n",avg_wt);
}
```

OUTPUT FOR ROUND ROBIN

```
sahal@kali:~/bash_script$ ./a.out

enter the number of processors
6

enter the burst time for 6 processors
5
6
3
1
5
4

enter the arrival time for 6 processors
0
1
2
3
4
```

```
6

***************************************************

processor       arrival time            burst time
1               0.000000                5.000000
2               1.000000                6.000000
3               2.000000                3.000000
4               3.000000                1.000000
5               4.000000                5.000000
6               6.000000                4.000000

***************************************************

enter the time quantum
4

turn around timie=15.833333

waiting time=11.833333
```