FCFS:

#include <stdio.h>

int waitingtime(int proc[], int n,

int burst\_time[], int wait\_time[]) {

wait\_time[0] = 0;

for (int i = 1; i < n ; i++ )

wait\_time[i] = burst\_time[i-1] + wait\_time[i-1] ;

return 0;

}

int turnaroundtime( int proc[], int n,

int burst\_time[], int wait\_time[], int tat[]) {

int i;

for ( i = 0; i < n ; i++)

tat[i] = burst\_time[i] + wait\_time[i];

return 0;

}

int avgtime( int proc[], int n, int burst\_time[]) {

int wait\_time[n], tat[n], total\_wt = 0, total\_tat = 0;

int i;

waitingtime(proc, n, burst\_time, wait\_time);

turnaroundtime(proc, n, burst\_time, wait\_time, tat);

printf("Processes Burst Waiting Turn around \n");

for ( i=0; i<n; i++) {

total\_wt = total\_wt + wait\_time[i];

total\_tat = total\_tat + tat[i];

printf(" %d\t %d\t\t %d \t%d\n", i+1, burst\_time[i], wait\_time[i], tat[i]);

}

printf("Average waiting time = %f\n", (float)total\_wt / (float)n);

printf("Average turn around time = %f\n", (float)total\_tat / (float)n);

return 0;

}

int main() {

int proc[] = { 1, 2, 3};

int n = sizeof proc / sizeof proc[0];

int burst\_time[] = {5, 8, 12};

avgtime(proc, n, burst\_time);

return 0;

}

Sjf:

#include<stdio.h>

**void** main()

{

**int** bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;

**float** avg\_wt,avg\_tat;

    printf("Enter number of process:");

    scanf("%d",&n);

    printf("\nEnter Burst Time:\n");

**for**(i=0;i<n;i++)

    {

        printf("p%d:",i+1);

        scanf("%d",&bt[i]);

        p[i]=i+1;           *//contains process number*

    }

*//sorting burst time in ascending order using selection sort*

**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];

        p[i]=p[pos];

        p[pos]=temp;

    }

    wt[0]=0;            *//waiting time for first process will be zero*

*//calculate waiting time*

**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;      *//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    %d\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);

}

ROUND ROBIN :

#include<stdio.h>

**int** main()

{

**int** count,j,n,time,remain,flag=0,time\_quantum;

**int** wait\_time=0,turnaround\_time=0,at[10],bt[10],rt[10];

**printf**("Enter Total Process:\t ");

  scanf("%d",&n);

  remain=n;

**for**(count=0;count<n;count++)

  {

**printf**("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);

    scanf("%d",&at[count]);

    scanf("%d",&bt[count]);

    rt[count]=bt[count];

  }

**printf**("Enter Time Quantum:\t");

  scanf("%d",&time\_quantum);

**printf**("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");

**for**(time=0,count=0;remain!=0;)

  {

**if**(rt[count]<=time\_quantum && rt[count]>0)

    {

      time+=rt[count];

      rt[count]=0;

      flag=1;

    }

**else** **if**(rt[count]>0)

    {

      rt[count]-=time\_quantum;

      time+=time\_quantum;

    }

**if**(rt[count]==0 && flag==1)

    {

      remain--;

**printf**("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);

      wait\_time+=time-at[count]-bt[count];

      turnaround\_time+=time-at[count];

      flag=0;

    }

**if**(count==n-1)

      count=0;

**else** **if**(at[count+1]<=time)

      count++;

**else**

      count=0;

  }

**printf**("\nAverage Waiting Time= %f\n",wait\_time\*1.0/n);

**printf**("Avg Turnaround Time = %f",turnaround\_time\*1.0/n);

**return** 0;

}

SRTF:

#include <stdio.h>

**int** main()

{

**int** arrival\_time[10], burst\_time[10], temp[10];

**int** i, smallest, count = 0, time, limit;

**double** wait\_time = 0, turnaround\_time = 0, end;

**float** average\_waiting\_time, average\_turnaround\_time;

      printf("nEnter the Total Number of Processes:t");

      scanf("%d", &limit);

      printf("nEnter Details of %d Processesn", limit);

**for**(i = 0; i < limit; i++)

      {

            printf("nEnter Arrival Time:t");

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

            printf("Enter Burst Time:t");

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

            temp[i] = burst\_time[i];

      }

      burst\_time[9] = 9999;

**for**(time = 0; count != limit; time++)

      {

            smallest = 9;

**for**(i = 0; i < limit; i++)

            {

**if**(arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

                  {

                        smallest = i;

                  }

            }

            burst\_time[smallest]--;

**if**(burst\_time[smallest] == 0)

            {

                  count++;

                  end = time + 1;

                  wait\_time = wait\_time + end - arrival\_time[smallest] - temp[smallest];

                  turnaround\_time = turnaround\_time + end - arrival\_time[smallest];

            }

      }

      average\_waiting\_time = wait\_time / limit;

      average\_turnaround\_time = turnaround\_time / limit;

      printf("nnAverage Waiting Time:t%lfn", average\_waiting\_time);

      printf("Average Turnaround Time:t%lfn", average\_turnaround\_time);

**return** 0;

}

PRIORITY:

|  |
| --- |
| #include<stdio.h>    **int** main()  {  **int** bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,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++)      {          wt[i]=0;  **for**(j=0;j<i;j++)              wt[i]+=bt[j];            total+=wt[i];      }        avg\_wt=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    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);      }        avg\_tat=total/n;     *//average turnaround time*  **printf**("\n\nAverage Waiting Time=%d",avg\_wt);  **printf**("\nAverage Turnaround Time=%d\n",avg\_tat);    **return** 0;  } |