

**lab Assignment**

**Operating Systems Lab**

**CSE 324**

**Submitted To:**

Most. Hasna Hena

Assistant Professor

Department of CSE

Daffodil International University

**Submitted By:**

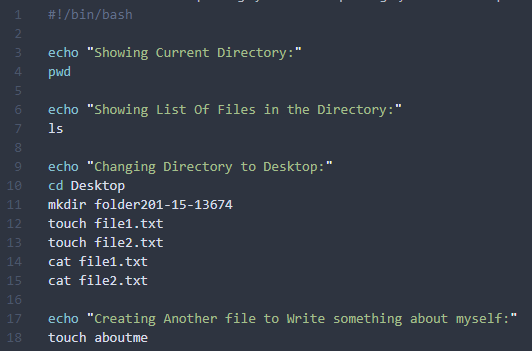
Fahim Ur Rahman

ID: 201-15-13674

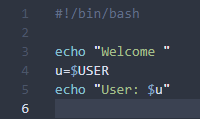
**Section:** B

**Submitted On:** 7th December, 2022

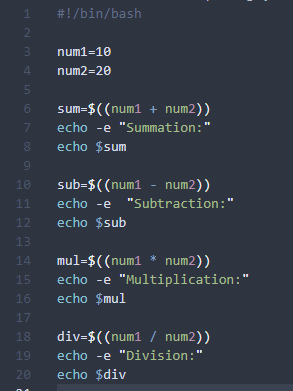
**Problem 1 Solution:**



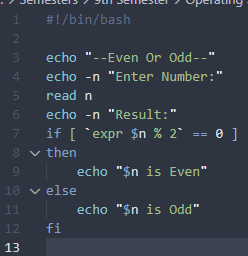
**Problem 2 Solution:**



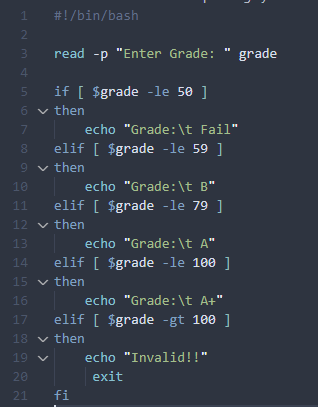
**Problem 3 Solution:**



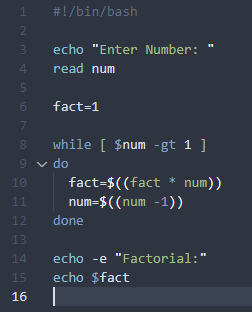
**Problem 4 Solution:**



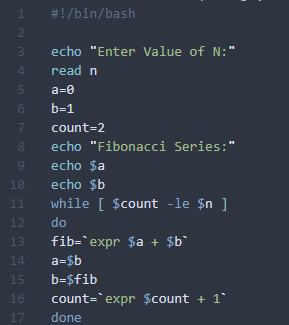
**Problem 5 Solution:**



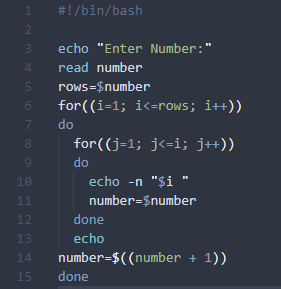
**Problem 6 Solution:**



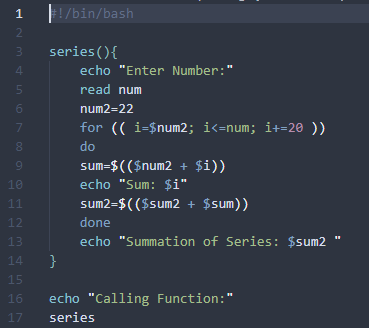
**Problem 7 Solution:**



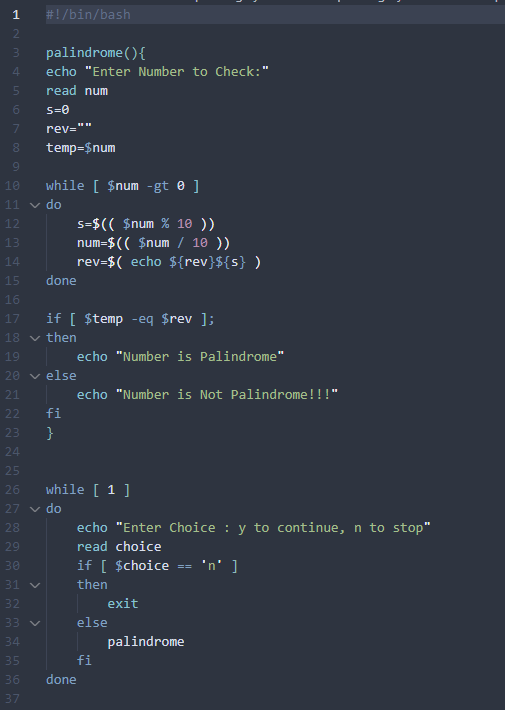
**Problem 8 Solution:**



**Problem 9 Solution:**



Problem 10 Solution:



**Problem 11 Solution:**

**#**include<stdio.h>

int main()

{

          int process[10],arrival[10],burst[10],start[10],end[10],wait[10],turnaround[10];

          int n,i,j,temp,totalwait=0,totalturn=0;

          float avgwait,avgturn;

          printf("Enter the number of processes\n");

          scanf("%d",&n);

          printf("Enter value for each process\n\n");

          printf("process name,arrival time,Burst time\n");

          for(i=0;i<n;i++)

          scanf("%d%d%d",&process[i],&arrival[i],&burst[i]);

          for(i=0;i<n-1;i++)

         {

                  for(j=i+1;j<n;j++)

                 {

                          if(arrival[i]>arrival[j])

                          {

                                     temp=arrival[i];

                                     arrival[i]=arrival[j];

                                     arrival[j]=temp;

                                     temp=process[i];

                                     process[i]=process[j];

                                     process[j]=temp;

                                     temp=burst[i];

                                     burst[i]=burst[j];

                                     burst[j]=temp;

                           }

                  }

          }

          printf("\nProcess name\tArrival time\tBurst time\n");

          for(i=0;i<n;i++)

          printf("p[%d]\t\t%d\t\t%d\n",process[i],arrival[i],burst[i]);

          start[0]=arrival[0];

          end[0]=arrival[0]+burst[0];

          wait[0]=0;

          turnaround[0]=burst[0];

          totalturn+=turnaround[0];

          for(i=1;i<n;i++)

         {

                   if(arrival[i]<=end[i-1])

                   start[i]=end[i-1];

                   else

                   start[i]=arrival[i];

                   wait[i]=start[i]-arrival[i];

                   totalwait+=wait[i];

                   end[i]=start[i]+burst[i];

                   turnaround[i]=burst[i]+wait[i];

                   totalturn+=turnaround[i];

          }

          avgwait=(float)totalwait/(float)n;

          avgturn=(float)totalturn/(float)n;

          printf("\n\n|Process name\t|Arrival time\t|Burst time\t|Start time\t|End time\t|Waiting time\t|Turnaround time|\n");

          for(i=0;i<n;i++)

          printf("|p[%d]\t\t|%d\t\t|%d\t\t|%d\t\t|%d\t\t|%d\t\t|%d\n",process[i],arrival[i],burst[i],start[i],

                       end[i],wait[i],turnaround[i]);

          printf("Total waiting time is %d\n",totalwait);

          printf("Average waiting time is %f\n",avgwait);

          printf("Total turnaround time is %d\n",totalturn);

          printf("Average turnaround time is %f\n",avgturn);

          return 0;

}

**Problem 12 Solution:**

**#**include<stdio.h>

int main()

{

            int at[10],bt[10],pr[10];

            int n,i,j,temp,time=0,count,over=0,sum\_wait=0,sum\_turnaround=0,start;

            float avgwait,avgturn;

            printf("Enter the number of processes\n");

            scanf("%d",&n);

            for(i=0;i<n;i++)

            {

                        printf("Enter the arrival time and burst time for process %d\n",i+1);

                        scanf("%d%d",&at[i],&bt[i]);

                        pr[i]=i+1;

            }

            for(i=0;i<n-1;i++)

            {

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

                        {

                                    if(at[i]>at[j])

                                    {

                                                temp=at[i];

                                                at[i]=at[j];

                                                at[j]=temp;

                                                temp=bt[i];

                                                bt[i]=bt[j];

                                                bt[j]=temp;

                                                temp=pr[i];

                                                pr[i]=pr[j];

                                                pr[j]=temp;

                                    }

                        }

            }

            printf("\n\nProcess\t|Arrival time\t|Burst time\t|Start time\t|End time\t|waiting time\t|Turnaround time\n\n");

            while(over<n)

            {

                        count=0;

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

                        {

                                    if(at[i]<=time)

                                    count++;

                                    else

                                    break;

                        }

                        if(count>1)

                        {

                                    for(i=over;i<over+count-1;i++)

                                    {

                                                for(j=i+1;j<over+count;j++)

                                                {

                                                            if(bt[i]>bt[j])

                                                            {

                                                                        temp=at[i];

                                                                        at[i]=at[j];

                                                                        at[j]=temp;

                                                                        temp=bt[i];

                                                                        bt[i]=bt[j];

                                                                        bt[j]=temp;

                                                                        temp=pr[i];

                                                                        pr[i]=pr[j];

                                                                        pr[j]=temp;

                                                            }

                                                }

                                    }

                        }

                        start=time;

                        time+=bt[over];

                         printf("p[%d]\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\n",pr[over],

                                    at[over],bt[over],start,time,time-at[over]-bt[over],time-at[over]);

                        sum\_wait+=time-at[over]-bt[over];

                        sum\_turnaround+=time-at[over];

                        over++;

            }

            avgwait=(float)sum\_wait/(float)n;

            avgturn=(float)sum\_turnaround/(float)n;

            printf("Average waiting time is %f\n",avgwait);

            printf("Average turnaround time is %f\n",avgturn);

            return 0;

}