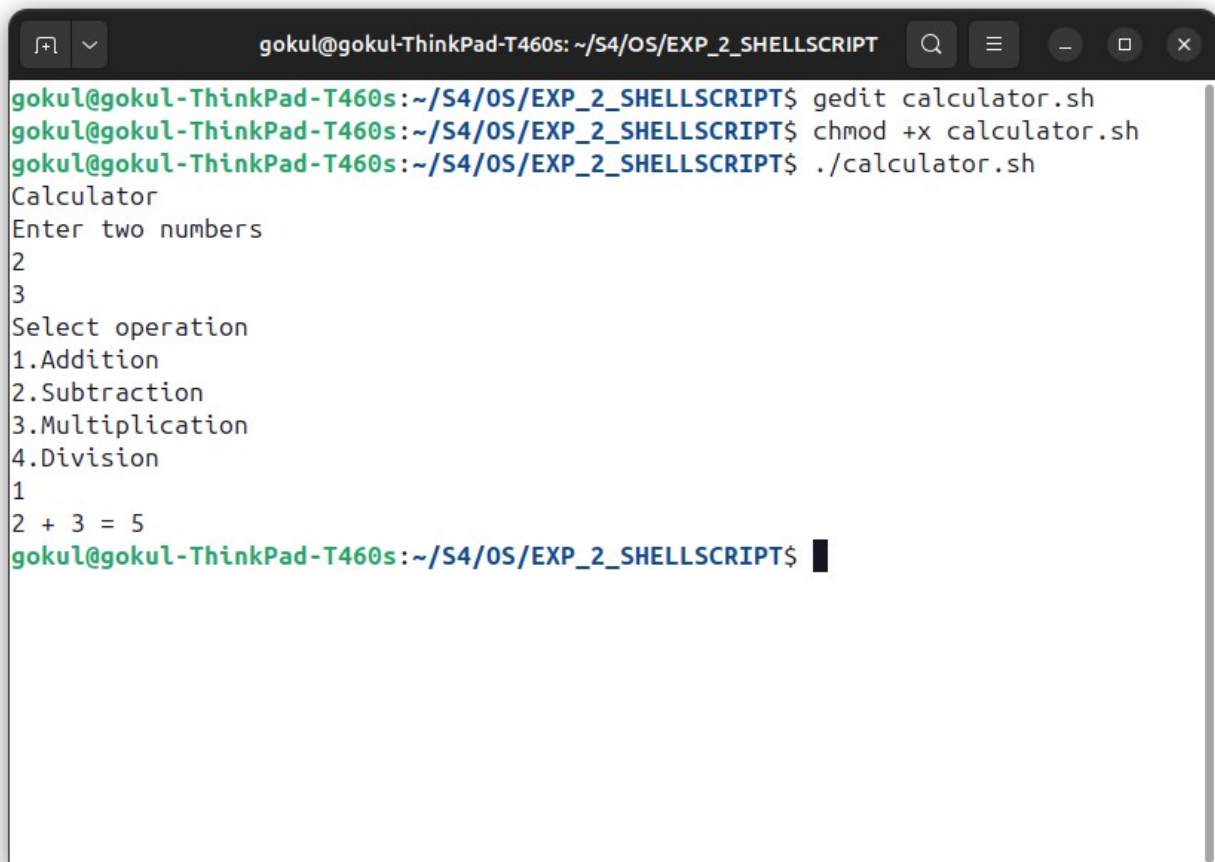


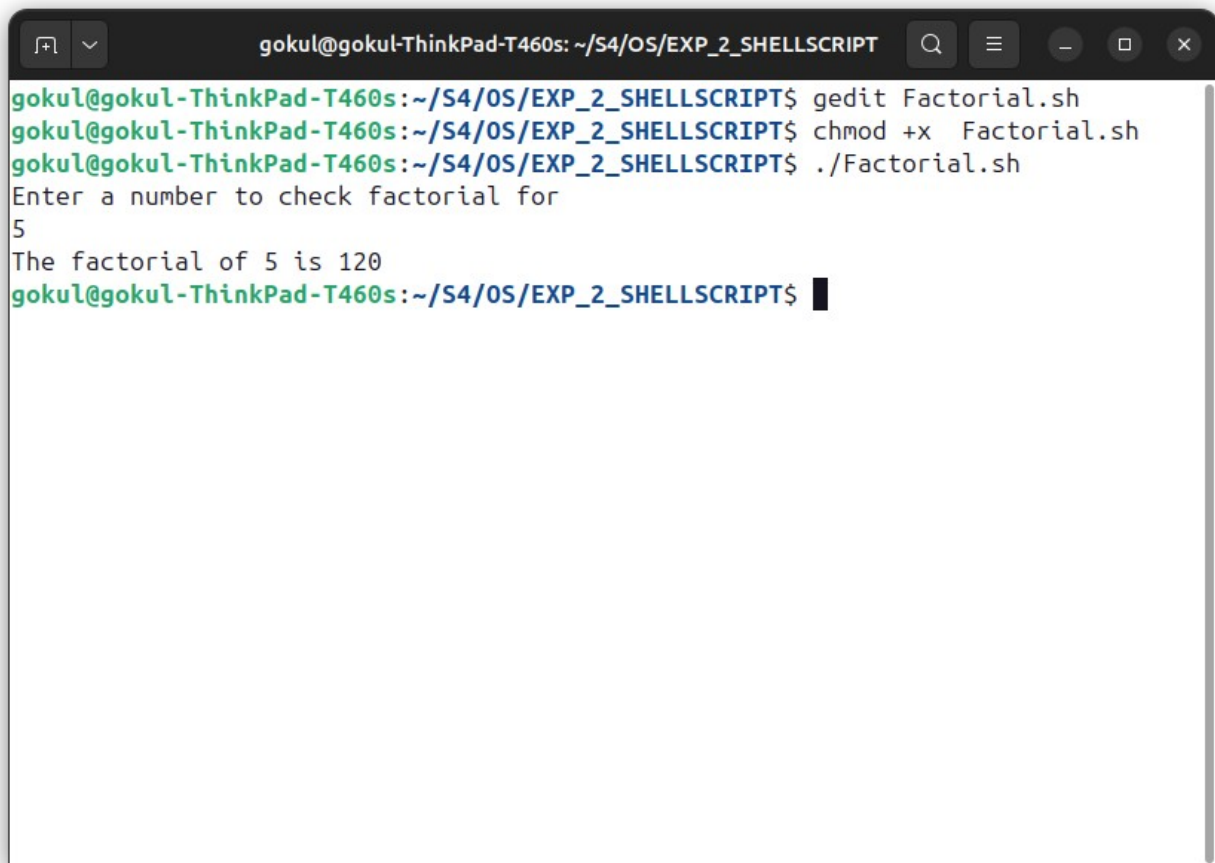
#Simple calculator using Switch

```
echo "Calculator"
echo "Enter two numbers"
read a
read b
echo "Select operation"
echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
read c
case $c in
1) echo "$a + $b = $((a+b))";;
2) echo "$a - $b = $((a-b))";;
3) echo "$a * $b = $((a*b))";;
4) echo "$a / $b = $((a/b))";;
esac
```

A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT' with standard window controls. The terminal shows the following sequence of commands and output:
1. Command: `gedit calculator.sh`
2. Command: `chmod +x calculator.sh`
3. Command: `./calculator.sh`
4. Output: `Calculator`
5. Prompt: `Enter two numbers`
6. Input: `2`
7. Input: `3`
8. Prompt: `Select operation`
9. Input: `1.Addition`
10. Output: `2 + 3 = 5`
11. Prompt: `gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$` with a cursor.

#Factorial of a number

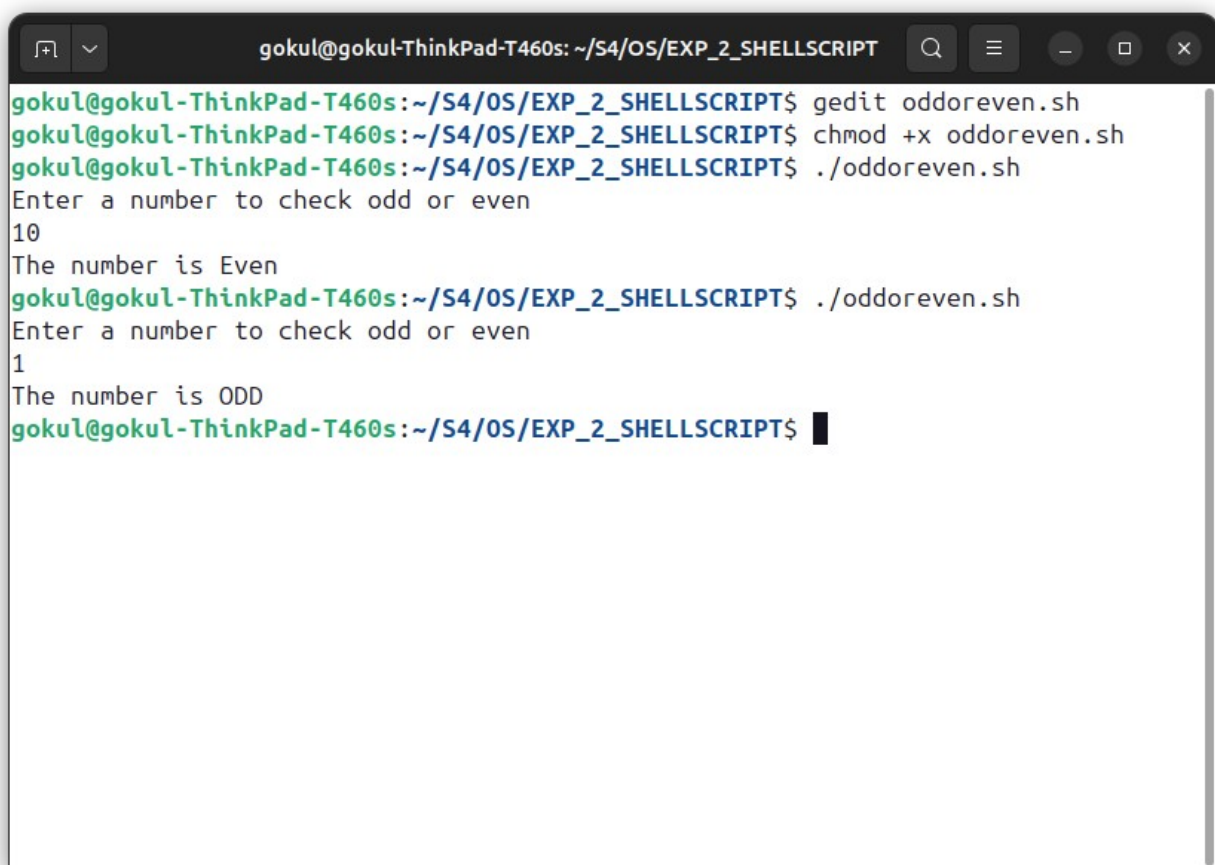
```
echo "Enter a number to check factorial for"
read a
num=$a
fact=1
while [ $a -gt 1 ];
do
    fact=$(( a * fact))
    a=$((a - 1))
done
echo "The factorial of $num is $fact"
```

A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT' with standard window controls. The terminal shows the following sequence of commands and output:
1. Command: `gedit Factorial.sh`
2. Command: `chmod +x Factorial.sh`
3. Command: `./Factorial.sh`
4. Prompt: `Enter a number to check factorial for`
5. Input: `5`
6. Output: `The factorial of 5 is 120`
7. Prompt: `gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT$` with a cursor.

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT$ gedit Factorial.sh
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT$ chmod +x Factorial.sh
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT$ ./Factorial.sh
Enter a number to check factorial for
5
The factorial of 5 is 120
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT$
```

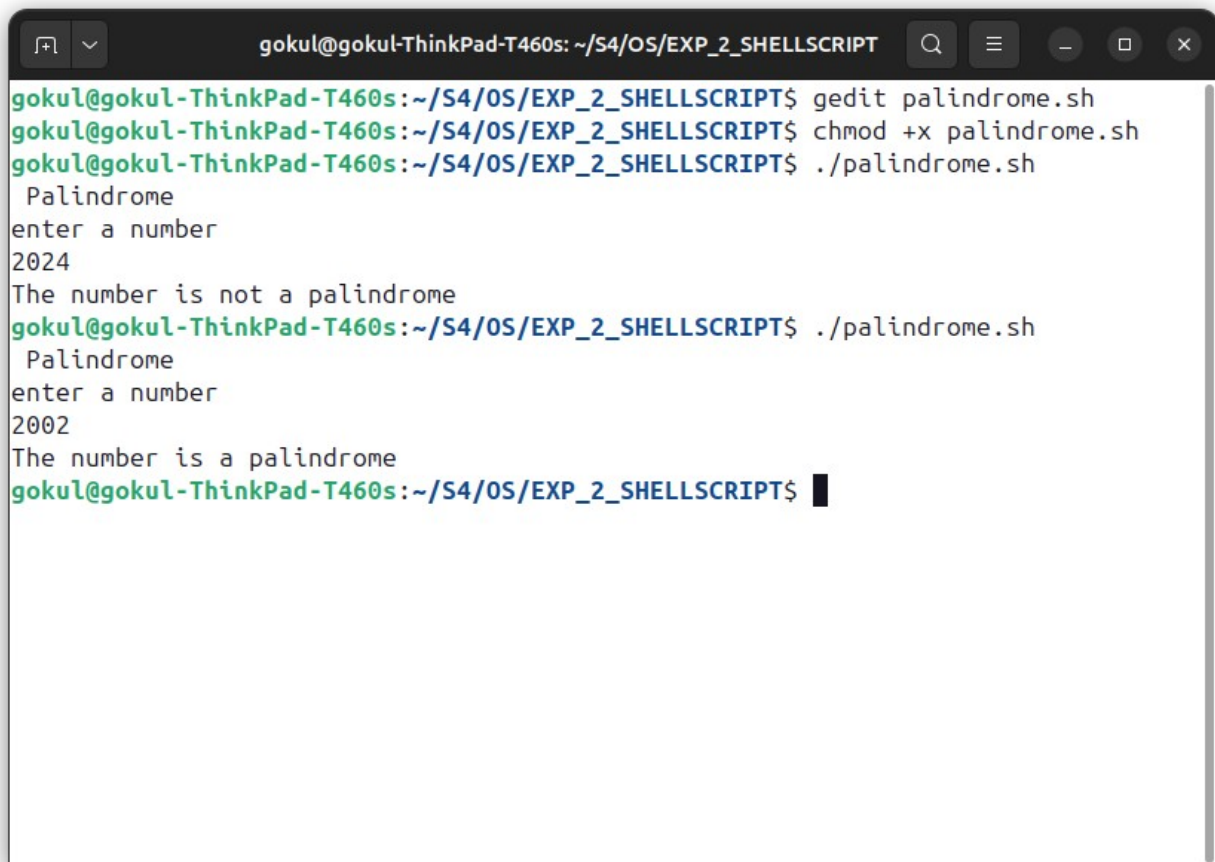
#Odd or Even

```
echo "Enter a number to check odd or even"
read a
n=$((a % 2))
if [ $n -eq 0 ]
then
    echo "The number is Even"
else
    echo "The number is ODD"
fi
```

A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT' with standard window controls. The terminal shows the following sequence of commands and outputs:
1. Command: `gedit oddoreven.sh`
2. Command: `chmod +x oddoreven.sh`
3. Command: `./oddoreven.sh`
 Output: `Enter a number to check odd or even`
 Input: `10`
 Output: `The number is Even`
4. Command: `./oddoreven.sh`
 Output: `Enter a number to check odd or even`
 Input: `1`
 Output: `The number is ODD`
5. The prompt `gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$` is shown with a cursor.

#Check Palindrome or not

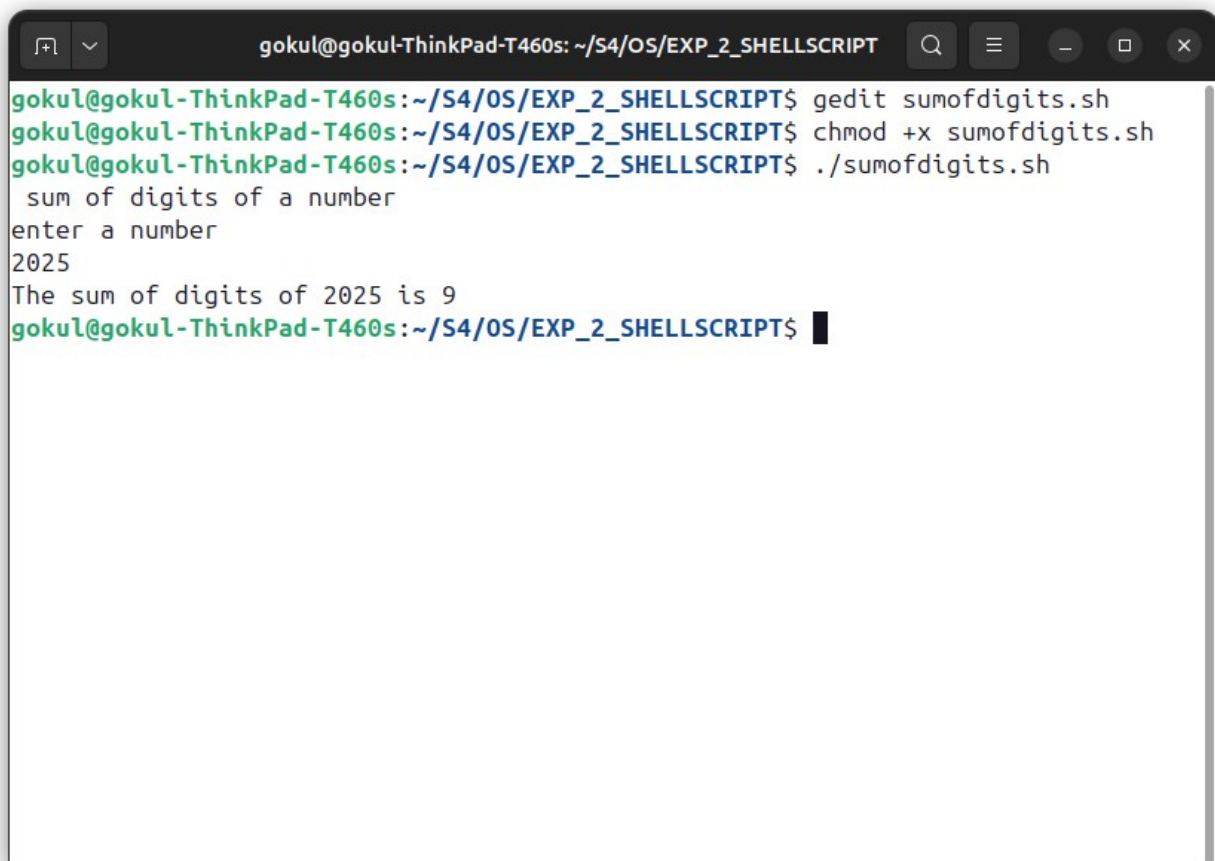
```
echo " Palindrome "  
echo "enter a number"  
read num  
no=$num  
rev=0  
while [ $num -gt 0 ]; do  
    digit=$(( num % 10 ))  
    rev=$(( rev * 10 + digit ))  
    num=$(( num / 10 ))  
done  
if [ $rev -eq $no ]  
then  
    echo "The number is a palindrome"  
else  
    echo "The number is not a palindrome"  
fi
```

A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT' showing the execution of a script named 'palindrome.sh'. The user runs 'gedit palindrome.sh', 'chmod +x palindrome.sh', and './palindrome.sh'. The script prompts for a number. The first input is '2024', and the output is 'The number is not a palindrome'. The second input is '2002', and the output is 'The number is a palindrome'.

```
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ gedit palindrome.sh  
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ chmod +x palindrome.sh  
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ ./palindrome.sh  
Palindrome  
enter a number  
2024  
The number is not a palindrome  
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ ./palindrome.sh  
Palindrome  
enter a number  
2002  
The number is a palindrome  
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$
```

#Calculate sum of digits of a number

```
echo "sum of digits of a number"
echo "enter a number"
read num
no=$num
sum=0
while [ $num -gt 0 ]; do
    digit=$(( num % 10 ))
    sum=$(( sum + digit ))
    num=$(( num / 10 ))
done
echo "The sum of digits of $no is $sum"
```

A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP_2_SHELLSCRIPT'. The window shows the following commands and output:

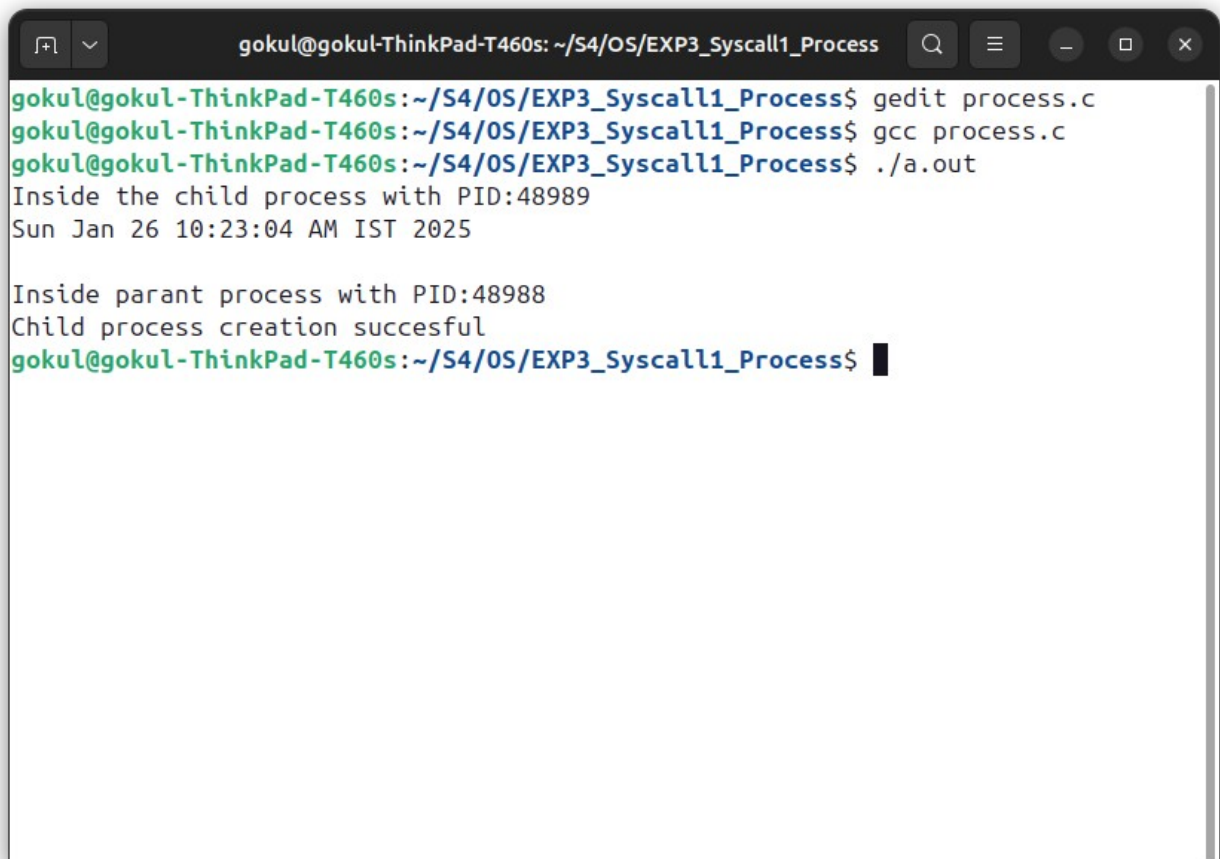
```
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ gedit sumofdigits.sh
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ chmod +x sumofdigits.sh
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$ ./sumofdigits.sh
sum of digits of a number
enter a number
2025
The sum of digits of 2025 is 9
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP_2_SHELLSCRIPT$
```

```

#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/wait.h>

int main(){
    int status,pid,child_pid;
    pid=fork();
    if(pid==-1)
    {
        printf("Child process creation failed\n");
        return -1;
    }
    else if(pid==0){
        printf("Inside the child process with PID:%d\n",getpid());
        execlp("/bin/date","date",NULL);
        return 0;
    }
    else{
        child_pid=wait(&status);
        printf("\nInside parant process with PID:%d\n",getpid());
        printf("Child process creation succesful\n");
        return 0;
    }
}

```



The image shows a terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP3_Syscall1_Process'. The user has executed the following commands: `gedit process.c`, `gcc process.c`, and `./a.out`. The output of the program is displayed in the terminal, showing the child process running 'date' and the parent process waiting for it to complete.

```

gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ gedit process.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ gcc process.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$ ./a.out
Inside the child process with PID:48989
Sun Jan 26 10:23:04 AM IST 2025

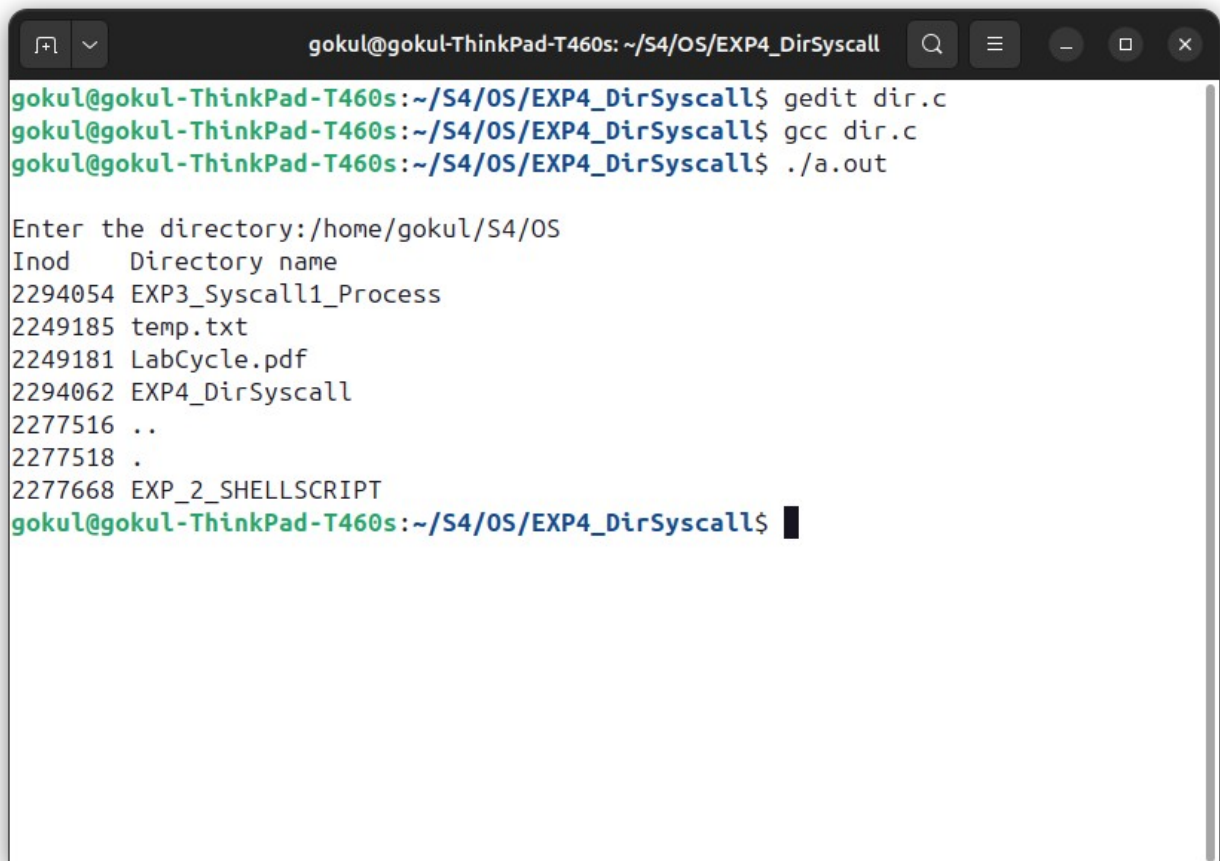
Inside parant process with PID:48988
Child process creation succesful
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP3_Syscall1_Process$

```



```
#include<stdio.h>
#include<sys/types.h>
#include<dirent.h>

void main(){
    DIR *dir;
    struct dirent *ptr;
    char dirname[100];
    printf("\nEnter the directory:");
    scanf("%s",dirname);
    dir=opendir(dirname);
    printf("Inod\tDirectory name\n");
    while((ptr=readdir(dir))!=NULL){
        printf("%ld\t%s\n",ptr->d_ino,ptr->d_name);
    }
    closedir(dir);
    return;
}
```



The image shows a terminal window titled "gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP4_DirSyscall". The user has executed the following commands: `gedit dir.c`, `gcc dir.c`, and `./a.out`. The program prompts "Enter the directory: /home/gokul/S4/OS" and then displays a list of files and directories with their inodes and names. The output is as follows:

```
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_DirSyscall$ gedit dir.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_DirSyscall$ gcc dir.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_DirSyscall$ ./a.out

Enter the directory:/home/gokul/S4/OS
Inod    Directory name
2294054 EXP3_Syscall1_Process
2249185 temp.txt
2249181 LabCycle.pdf
2294062 EXP4_DirSyscall
2277516 ..
2277518 .
2277668 EXP_2_SHELLSCRIPT
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_DirSyscall$
```

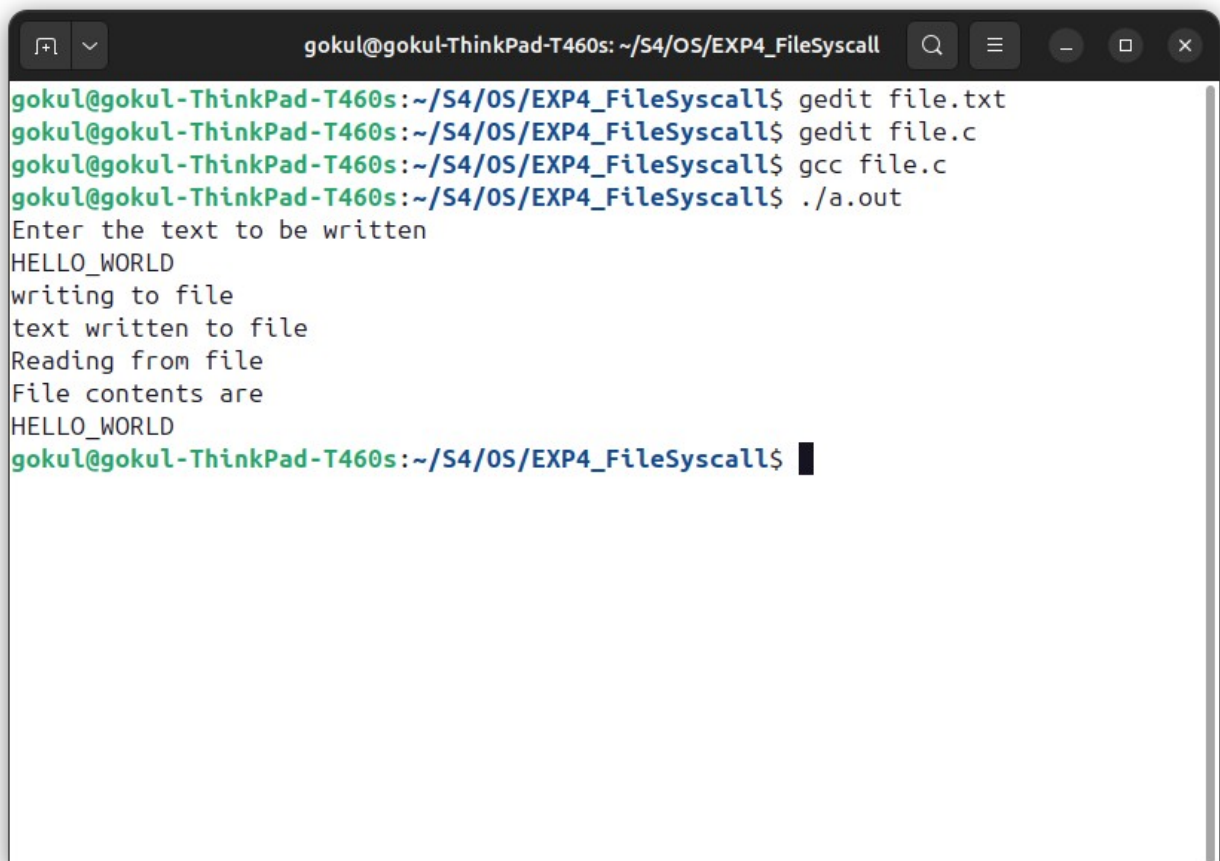


```

#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/stat.h>
#include<fcntl.h>

int main(){
    int fd,fd2;
    char wbuf[128],rbuf[128];
    fd=open("file.txt",O_WRONLY);
    printf("Enter the text to be written\n");
    scanf("%s",wbuf);
    printf("writing to file\n");
    write(fd,wbuf,strlen(wbuf));
    close(fd);
    printf("text written to file\n");
    printf("Reading from file\n");
    fd2=open("file.txt",O_RDONLY);
    printf("File contents are\n");
    read(fd2,rbuf,128);
    printf("%s\n",rbuf);
    close(fd2);
    return 0;
}

```



A terminal window titled 'gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP4_FileSyscall' showing the execution of a C program. The user enters 'HELLO_WORLD' and the program successfully writes it to 'file.txt' and reads it back, displaying 'HELLO_WORLD'.

```

gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_FileSyscall$ gedit file.txt
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_FileSyscall$ gedit file.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_FileSyscall$ gcc file.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_FileSyscall$ ./a.out
Enter the text to be written
HELLO_WORLD
writing to file
text written to file
Reading from file
File contents are
HELLO_WORLD
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP4_FileSyscall$

```


1) First Come First Serve

```
#include<stdio.h>
int main(){
    int p[20],bt[20],wt[20],tt[20],n,i;
    float avg_wt=0.0,avg_tt=0.0;
    printf("Enter the number of processes: ");
    scanf("%d",&n);
    printf("Enter the  and process id:\n");
    for(i=0;i<n;i++){
        printf("For process %d: ",i+1);
        scanf("%d",&p[i]);
    }
    printf("Enter the burst time:\n");
    for(i=0;i<n;i++){
        printf("For process P%d: ",p[i]);
        scanf("%d",&bt[i]);
    }
    wt[0]=0;
    for(i=0;i<n;i++){
        wt[i]=wt[i-1]+bt[i-1];
        avg_wt+=wt[i];
    }
    avg_wt=avg_wt/n;
    for(i=0;i<n;i++){
        tt[i]=wt[i]+bt[i];
        avg_tt+=tt[i];
    }
    avg_tt=avg_tt/n;
    printf("_____ \n");
n");
    printf("\tPID\t\tBT\t\tWT\t\tTT\t\n");
    for(i=0;i<n;i++){
        printf("\t%d\t\t%d\t\t%d\t\t%d\t\n",p[i],bt[i],wt[i],tt[i]);
    }
    printf("_____ \n");
n");
    printf("GANTT CHART");
    printf("\n_____ \n");
n");
    for(i=0;i<n;i++)
    {
        printf("|P%d\t|",p[i]);
    }
```

```

printf("\n_____ \n");
printf("%d\t",wt[0]);
for(i=0;i<n;i++){
    printf("%d\t",tt[i]);
}
printf("\nThe average waiting time is:%4f\n",avg_wt);
printf("\nThe average turn around time is:%4f\n",avg_tt);
}

```

```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP6_NonPreemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gedit FCFS.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gcc FCFS.c -o FCFS.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ ./FCFS.out
Enter the number of processes: 3
Enter the and process id:
For process 1: 1
For process 2: 2
For process 3: 3
Enter the burst time:
For process P1: 2
For process P2: 3
For process P3: 4

```

| PID | BT | WT | TT |
|-----|----|----|----|
| 1 | 2 | 0 | 2 |
| 2 | 3 | 2 | 5 |
| 3 | 4 | 5 | 9 |

```

GANTT CHART
|P1  ||P2  ||P3  |
0    2    5    9
The average waiting time is:2.333333
The average turn around time is:5.333333
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$

```

2)Shortest Job First

```
#include<stdio.h>
int main(){
    int p[20],bt[20],wt[20],tt[20],n,i,j,temp;
    float avg_wt=0.0,avg_tt=0.0;
    printf("Enter the number of processes: ");
    scanf("%d",&n);
    printf("Enter the  and process id:\n");
    for(i=0;i<n;i++){
        printf("For process %d: ",i+1);
        scanf("%d",&p[i]);
    }
    printf("Enter the burst time:\n");
    for(i=0;i<n;i++){
        printf("For process P%d: ",p[i]);
        scanf("%d",&bt[i]);
    }
    for(i=0;i<n-1;i++){
        for(j=0;j<n-i-1;j++){
            if(bt[j]>bt[j+1]){
                temp=bt[j];
                bt[j]=bt[j+1];
                bt[j+1]=temp;
                temp=p[j];
                p[j]=p[j+1];
                p[j+1]=temp;
            }
        }
    }
    wt[0]=0;
    for(i=0;i<n;i++){
        wt[i]=wt[i-1]+bt[i-1];
        avg_wt+=wt[i];
    }
    avg_wt=avg_wt/n;
    for(i=0;i<n;i++){
        tt[i]=wt[i]+bt[i];
        avg_tt+=tt[i];
    }
    avg_tt=avg_tt/n;
```

```

printf("_____\\n");
printf("\\tPID\\t\\tBT\\t\\tWT\\t\\tTT\\t\\n");
for(i=0;i<n;i++){
    printf("\\t%d\\t\\t%d\\t\\t%d\\t\\t%d\\t\\n",p[i],bt[i],wt[i],tt[i]);
}
printf("_____\\
n");
printf("GANTT CHART");
printf("\\n_____\\
n");
for(i=0;i<n;i++)
{
    printf("|P%d\\t|",p[i]);
}
printf("\\n_____\\n");
printf("%d\\t",wt[0]);
for(i=0;i<n;i++){
    printf("%d\\t",tt[i]);
}
printf("\\nThe average waiting time is:%4f\\n",avg_wt);
printf("\\nThe average turn around time is:%4f\\n",avg_tt);
}

```

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP6_NonPreemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gedit SJF.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gcc SJF.c -o SJF.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ ./SJF.out
Enter the number of processes: 3
Enter the and process id:
For process 1: 1
For process 2: 2
For process 3: 3
Enter the burst time:
For process P1: 6
For process P2: 3
For process P3: 9
```

| PID | BT | WT | TT |
|-----|----|----|----|
| 2 | 3 | 0 | 3 |
| 1 | 6 | 3 | 9 |
| 3 | 9 | 9 | 18 |

GANTT CHART

| P2 | P1 | P3 | |
|----|----|----|----|
| 0 | 3 | 9 | 18 |

```
The average waiting time is:4.000000
The average turn around time is:10.000000
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$
```

3) Priority

```
#include<stdio.h>
int main(){
    int p[20],bt[20],wt[20],tt[20],n,i,j,temp,pr[20];
    float avg_wt=0.0,avg_tt=0.0;
    printf("Enter the number of processes: ");
    scanf("%d",&n);
    printf("Enter the  and process id:\n");
    for(i=0;i<n;i++){
        printf("For process %d: ",i+1);
        scanf("%d",&p[i]);
    }
    printf("Enter the burst time:\n");
    for(i=0;i<n;i++){
        printf("For process P%d: ",p[i]);
        scanf("%d",&bt[i]);
    }
    printf("Enter the Priority:\n");
    for(i=0;i<n;i++){
        printf("For process P%d: ",p[i]);
        scanf("%d",&pr[i]);
    }
    for(i=0;i<n-1;i++){
        for(j=0;j<n-i-1;j++){
            if(pr[j]>pr[j+1]){
                temp=pr[j];
                pr[j]=pr[j+1];
                pr[j+1]=temp;
                temp=bt[j];
                bt[j]=bt[j+1];
                bt[j+1]=temp;
                temp=p[j];
                p[j]=p[j+1];
                p[j+1]=temp;
            }
        }
    }
    wt[0]=0;
    for(i=1;i<n;i++){
        wt[i]=wt[i-1]+bt[i-1];
        avg_wt+=wt[i];
    }
    avg_wt=avg_wt/n;
```



```

for(i=0;i<n;i++){
    tt[i]=wt[i]+bt[i];
    avg_tt+=tt[i];
}
avg_tt=avg_tt/n;
printf("_____ \
n");
printf("\tPID\t\tBT\t\tWT\t\tTT\t\n");
for(i=0;i<n;i++){
    printf("\t%d\t\t%d\t\t%d\t\t%d\t\n",p[i],bt[i],wt[i],tt[i]);
}
printf("_____ \
n");
printf("GANTT CHART");
printf("\n_____ \
n");
for(i=0;i<n;i++)
{
    printf("P%d\t",p[i]);
}
printf("\n_____ \n");
printf("%d\t",wt[0]);
for(i=0;i<n;i++){
    printf("%d\t",tt[i]);
}
printf("\nThe average waiting time is:%4f\n",avg_wt);
printf("The average turn around time is:%4f\n",avg_tt);
}

```

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP6_NonPreemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gedit Priority.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ gcc Priority.c -o Priority.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$ ./Priority.out
Enter the number of processes: 3
Enter the and process id:
For process 1: 1
For process 2: 2
For process 3: 3
Enter the burst time:
For process P1: 2
For process P2: 3
For process P3: 4
Enter the Priority:
For process P1: 8
For process P2: 4
For process P3: 1
```

| PID | BT | WT | TT |
|-----|----|----|----|
| 3 | 4 | 0 | 4 |
| 2 | 3 | 4 | 7 |
| 1 | 2 | 7 | 9 |

GANTT CHART

| P3 | P2 | P1 |
|----|----|----|
| 0 | 4 | 7 |
| | | 9 |

```
The average waiting time is:3.666667
The average turn around time is:6.666667
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP6_NonPreemptive$
```

1)Shortest Remaining Time First

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int n,at[10],bt[10],temp[10],ct[10],wt[10],i,j,time=0,count=0,tat[10],smallest;
```

```
    float avg_tat,avg_wt,end=0;
```

```
    printf("Enter the number of process:");
```

```
    scanf("%d",&n);
```

```
    printf("Enter the Process Details\n");
```

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

```
        printf("Enter the values for process%d:\n",i+1);
```

```
        printf("Arrival Time: ");
```

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

```
        printf("Burst Time: ");
```

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

```
        temp[i]=bt[i];
```

```
    }
```

```
    bt[9]=1e9;
```

```
    for(time=0;count!=n;time++){
```

```
        smallest=9;
```

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

```
        if(at[i]<=time&&bt[i]<bt[smallest]&&bt[i]>0){
```

```
            smallest=i;
```

```
        }
```

```
    }
```

```
    if(smallest!=9){
```

```
        bt[smallest]--;
```

```
        if(bt[smallest]==0)
```

```
        {
```

```
            count++;
```

```
            end=time+1;
```

```
            ct[smallest]=end;
```

```
            tat[smallest]=end-at[smallest];
```

```
            wt[smallest]=end-at[smallest]-temp[smallest];
```

```
        }
```

```
    }
```

```
}
```

```
printf("_____\n");
```

```
printf("\tPID\t\tAT\t\tBT\t\tCT\t\tTT\t\tWT\t\t\n");
```

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

```
    printf("\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t\n",i+1,at[i],temp[i],ct[i],tat[i],wt[i]);
```

```
}
```

```

printf("_____
_____\\n");
    for(i=0;i<n;i++){
        avg_tat+=tat[i];
        avg_wt+=wt[i];
    }
    printf("the average turn around time is:%4f\\n",avg_tat/n);
    printf("the average waiting time is:%4f\\n",avg_wt/n);
}

```

```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP7_Preemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gedit SRTF.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gcc SRTF.c -o SRTF.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ ./SRTF.out
Enter the number of process:3
Enter the Process Details
Enter the values for process1:
Arrival Time: 0
Burst Time: 5
Enter the values for process2:
Arrival Time: 2
Burst Time: 2
Enter the values for process3:
Arrival Time: 3
Burst Time: 1

```

| PID | AT | BT | CT | TT | WT |
|-----|----|----|----|----|----|
| 1 | 0 | 5 | 8 | 8 | 3 |
| 2 | 2 | 2 | 4 | 2 | 0 |
| 3 | 3 | 1 | 5 | 2 | 1 |

```

the average turn around time is:4.000000
the average waiting time is:1.333333
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$

```

2)Priority

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int n,at[10],bt[10],temp[10],ct[10],wt[10],i,j,time=0,count=0,tat[10],smallest,pr[10];
```

```
    float avg_tat=0,avg_wt=0,end=0;
```

```
    printf("Enter the number of process:");
```

```
    scanf("%d",&n);
```

```
    printf("Enter the Process Details\n");
```

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

```
        printf("Enter the values for process%d:\n",i+1);
```

```
        printf("Arrival Time: ");
```

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

```
        printf("Burst Time: ");
```

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

```
        printf("Priority: ");
```

```
        scanf("%d",&pr[i]);
```

```
        temp[i]=bt[i];
```

```
    }
```

```
    pr[9]=1e9;
```

```
    for(time=0;count!=n;time++){
```

```
        smallest=9;
```

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

```
        if(at[i]<=time&&pr[i]<pr[smallest]&&bt[i]>0){
```

```
            smallest=i;
```

```
        }
```

```
    }
```

```
    if(smallest!=9){
```

```
        bt[smallest]--;
```

```
        if(bt[smallest]==0)
```

```
        {
```

```
            count++;
```

```
            end=time+1;
```

```
            ct[smallest]=end;
```

```
            tat[smallest]=end-at[smallest];
```

```
            wt[smallest]=end-at[smallest]-temp[smallest];
```

```
        }
```

```
    }
```

```
}
```

```
printf("_____\n");
```

```
printf("\tPID\t\tAT\t\tBT\t\tCT\t\tTT\t\tWT\n");
```

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

```
    printf("\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\n",i+1,at[i],temp[i],ct[i],tat[i],wt[i]);
```

```
}
```

```

printf("_____
_____\\n");
    for(i=0;i<n;i++){
        avg_tat+=tat[i];
        avg_wt+=wt[i];
    }
    printf("the average turn around time is:%4f\\n",avg_tat/n);
    printf("the average waiting time is:%4f\\n",avg_wt/n);
}

```

```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP7_Preemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gedit pr.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gcc pr.c -o Priority.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ ./Priority.out
Enter the number of process:3
Enter the Process Details
Enter the values for process1:
Arrival Time: 0
Burst Time: 5
Priority: 9
Enter the values for process2:
Arrival Time: 2
Burst Time: 2
Priority: 2
Enter the values for process3:
Arrival Time: 2
Burst Time: 3
Priority: 3

```

| | PID | AT | BT | CT | TT | WT |
|--|-----|----|----|----|----|----|
| | 1 | 0 | 5 | 10 | 10 | 5 |
| | 2 | 2 | 2 | 4 | 2 | 0 |
| | 3 | 2 | 3 | 7 | 5 | 2 |

```

the average turn around time is:5.666667
the average waiting time is:2.333333
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$

```

3) Round Robin

```
#include <stdio.h>

void main() {
    int i, j, n, qt, count = 0, time = 0, index = 0;
    int bt[20], bt_cp[20], p[20], ct[20], wt[20], tt[20];
    int gantt_process[100], gantt_time[100];
    float wt_avg = 0, tt_avg = 0;

    printf("Enter the number of Processes (Max 20): ");
    scanf("%d", &n);

    printf("Enter the Burst Time of Each Process:\n");
    for (i = 0; i < n; i++) {
        p[i] = i + 1;
        printf("P%d : ", p[i]);
        scanf("%d", &bt[i]);
        bt_cp[i] = bt[i];
        ct[i] = 0;
    }

    printf("Enter the Time Slice: ");
    scanf("%d", &qt);

    while (count != n) {
        for (i = 0; i < n; i++) {
            if (bt_cp[i] == 0) {
                continue;
            }
            if (bt_cp[i] > qt) {
                gantt_process[index] = p[i];
                gantt_time[index] = time + qt;
                time += qt;
                bt_cp[i] -= qt;
            } else {
                gantt_process[index] = p[i];
                gantt_time[index] = time + bt_cp[i];
                time += bt_cp[i];
                bt_cp[i] = 0;
                ct[i] = time;
                count++;
            }
            index++;
        }
    }

    for (i = 0; i < n; i++) {
        tt[i] = ct[i];
        wt[i] = tt[i] - bt[i];
        wt_avg += wt[i];
        tt_avg += tt[i];
    }
}
```

```

}

wt_avg /= n;
tt_avg /= n;

printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time\tCompletion Time\n");
for (i = 0; i < n; i++) {
    printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", p[i], bt[i], wt[i], tt[i], ct[i]);
}

printf("\nAverage Waiting Time: %.2f", wt_avg);
printf("\nAverage Turnaround Time: %.2f\n", tt_avg);

printf("\nGantt Chart:\n");
for (i = 0; i < index; i++) {
    printf("| P%d ", gantt_process[i]);
}
printf("\n");
printf("0");
for (i = 0; i < index; i++) {
    printf("   %d", gantt_time[i]);
}
printf("\n");
}

```

```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/LabCycle/EXP7_Preemptive
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gedit rr.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gcc rr.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ gcc rr.c -o rr.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$ ./rr.out
Enter the number of Processes (Max 20): 3
Enter the Burst Time of Each Process:
P1 : 2
P2 : 3
P3 : 1
Enter the Time Slice: 2

Process      Burst Time    Waiting Time    Turnaround Time    Completion Time
P1           2             0              2                  2
P2           3             3              6                  6
P3           1             4              5                  5

Average Waiting Time: 2.33
Average Turnaround Time: 4.33

Gantt Chart:
| P1 | P2 | P3 | P2 |
0   2   4   5   6
gokul@gokul-ThinkPad-T460s:~/S4/OS/LabCycle/EXP7_Preemptive$

```


Writer process:

```
#include<stdio.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>
#include<unistd.h>

int main(){
    int shid,nt,i;
    void *shmp;
    char buffer[100];

    int avail;
    shid=shmget(01234,sizeof(shmp),0644|IPC_CREAT);// create a shared memory area
    if(shid==-1){
        printf("Failed to create shared memory pool aborting process\n");
        return 1;
    }
    printf("Shared Memory Area Created\n");

    shmp=shmat(shid,NULL,0);//attach to the shared memory
    if(shmp==(void *)-1){
        printf("Failed to attach to shared memory,process terminating\n");
        return 1;
    }
    printf("Writer: Attached to the Shared Memory Area\n");

    //writing to shared memory
    printf("Enter the data to be written:\t");
    scanf("%s",buffer);
    strcpy(shmp,buffer);

    printf("Writing Process:Writing finished\n");

    if(shmdt(shmp)==-1){//detaching from shared memory
        printf("Failed to detach from Shared memory, process terminating\n");
        return 1;
    }
    printf("Writer:Detached from shared memory\n");

    printf("Writing process Completed\n");
    return 0;
}
```

Reader process:

```
#include<stdio.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>
#include<unistd.h>

int main(){
    int shid,nt,i;
    void *shmp;
    char buffer[100];

    int avail;
    shid=shmget(01234,sizeof(shmp),0644|IPC_CREAT);// create a shared memory area
    if(shid===-1){
        printf("Failed to create shared memory pool aborting process\n");
        return 1;
    }

    shmp=shmat(shid,NULL,0);//attach to the shared memory
    if(shmp==(void *)-1){
        printf("Failed to attach to shared memory,process terminating\n");
        return 1;
    }
    printf("Reader:Attached to Shared memory\n");

    //reading from shm

    printf("The Contents Read from Shared memory:\t");
    strcpy(buffer,shmp);
    printf("%s\n",buffer);

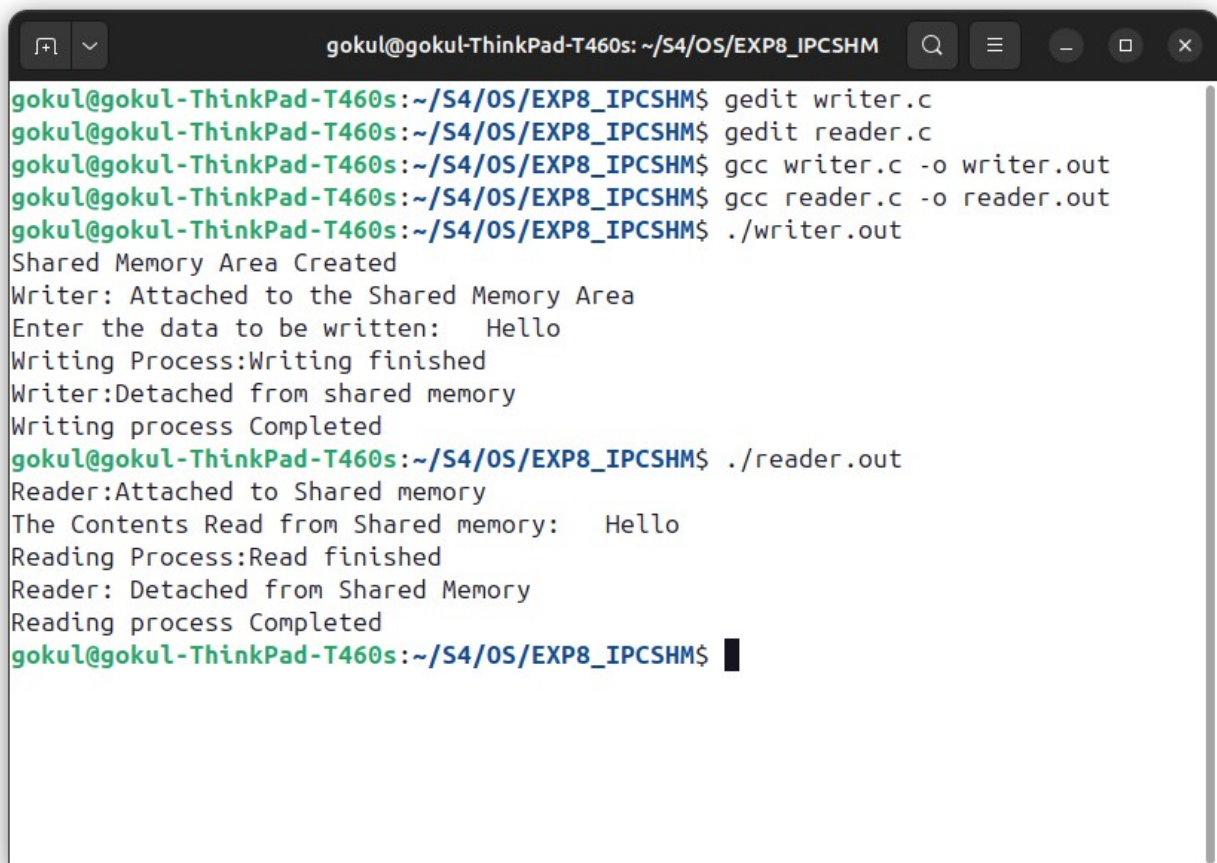
    printf("Reading Process:Read finished\n");

    if(shmdt(shmp)==-1){//detaching from shared memory
        printf("Failed to detach from Shared memory, process terminating\n");
        return 1;
    }
    printf("Reader: Detached from Shared Memory\n");
```

```

    if(shmctl(shid, IPC_RMID, 0) == -1) { //Destroying the shared memory
        printf("Failed to Destroy shared memory, Process terminating\n");
        return 1;
    }
    printf("Reading process Completed\n");
    return 0;
}

```



```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP8_IPCSHM
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ gedit writer.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ gedit reader.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ gcc writer.c -o writer.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ gcc reader.c -o reader.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ ./writer.out
Shared Memory Area Created
Writer: Attached to the Shared Memory Area
Enter the data to be written:  Hello
Writing Process:Writing finished
Writer:Detached from shared memory
Writing process Completed
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$ ./reader.out
Reader:Attached to Shared memory
The Contents Read from Shared memory:  Hello
Reading Process:Read finished
Reader: Detached from Shared Memory
Reading process Completed
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP8_IPCSHM$

```

Bankers Algorithm

program

```
#include<stdio.h>
#include<stdlib.h>

typedef struct {
    int res[10];
    int max[10][10];
    int alloc[10][10];
    int need[10][10];
    int avail[10];
} bankers;

bankers input(bankers b, int p, int r) {
    int i, j;
    printf("Enter the Number of Total resources in the system:\n");
    for (i = 0; i < r; i++) {
        printf("for resource R%d: ", i);
        scanf("%d", &b.res[i]);
    }

    printf("Enter the resources allocated to each process (allocated resource table):\n");
    for (i = 0; i < p; i++) {
        //printf("For Process P%d:\n", i);
        for (j = 0; j < r; j++) {
            //printf("for resource R%d: ", j);
            scanf("%d", &b.alloc[i][j]);
        }
    }

    printf("Enter the Maximum resources needed by each process (Max claim table):\n");
    for (i = 0; i < p; i++) {
        //printf("For Process P%d:\n", i);
        for (j = 0; j < r; j++) {
            //printf("for resource R%d: ", j);
            scanf("%d", &b.max[i][j]);
        }
    }

    for (i = 0; i < p; i++) {
        for (j = 0; j < r; j++) {
            b.need[i][j] = b.max[i][j] - b.alloc[i][j];
        }
    }

    for (i = 0; i < r; i++) {
        b.avail[i] = b.res[i];
        for (j = 0; j < p; j++) {
            b.avail[i] -= b.alloc[j][i];
        }
    }
}
```

```

    }

    return b;
}

int safety(bankers b, int p, int r) {
    int i, j, flag = 0, target = 0;
    int finish[10] = {0};
    int work[10];
    int safeSequence[10];

    for (i = 0; i < r; i++) {
        work[i] = b.avail[i];
    }

    while (target < p) {
        flag = 0;
        for (i = 0; i < p; i++) {
            if (!finish[i]) {
                int can_allocate = 1;
                for (j = 0; j < r; j++) {
                    if (b.need[i][j] > work[j]) {
                        can_allocate = 0;
                        break;
                    }
                }

                if (can_allocate) {
                    finish[i] = 1;
                    safeSequence[target++] = i;
                    for (j = 0; j < r; j++) {
                        work[j] += b.alloc[i][j];
                    }
                    flag = 1;
                    break;
                }
            }
        }

        if (!flag) {
            printf("System is in an unsafe state! Deadlock may occur.\n");
            return 0;
        }
    }

    printf("System is in a safe state! Safe Sequence: ");
    for (i = 0; i < p; i++) {
        printf("P%d ", safeSequence[i]);
    }
    printf("\n");
    return 1;
}

```

```

bankers request(bankers b, int p, int r) {
    int process_id, i, request[10];

    printf("Enter the process number (0 to %d): ", p - 1);
    scanf("%d", &process_id);

    if (process_id < 0 || process_id >= p) {
        printf("Invalid process ID!\n");
        return b;
    }

    printf("Enter the resource request for process P%d:\n", process_id);
    for (i = 0; i < r; i++) {
        printf("Resource R%d: ", i);
        scanf("%d", &request[i]);
    }

    for (i = 0; i < r; i++) {
        if (request[i] > b.need[process_id][i]) {
            printf("Error: Process has exceeded its maximum claim!\n");
            return b;
        }
        if (request[i] > b.avail[i]) {
            printf("Error: Insufficient resources available!\n");
            return b;
        }
    }

    for (i = 0; i < r; i++) {
        b.avail[i] -= request[i];
        b.alloc[process_id][i] += request[i];
        b.need[process_id][i] -= request[i];
    }

    if (safety(b, p, r)) {
        printf("Resource request granted!\n");
    } else {
        for (i = 0; i < r; i++) {
            b.avail[i] += request[i];
            b.alloc[process_id][i] -= request[i];
            b.need[process_id][i] += request[i];
        }
        printf("Resource request denied! System would enter an unsafe state.\n");
    }

    return b;
}

int main() {
    bankers b;

```

```

int p, r, choice;

printf("Enter number of processes: ");
scanf("%d", &p);
printf("Enter number of resources: ");
scanf("%d", &r);

b = input(b, p, r);

while (1) {
    printf("\nMenu:\n");
    printf("1. Check Safe State (Banker's Algorithm)\n");
    printf("2. Request Resources\n");
    printf("3. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            safety(b, p, r);
            break;
        case 2:
            b = request(b, p, r);
            break;
        case 3:
            printf("Exiting the program.\n");
            exit(0);
        default:
            printf("Invalid choice! Please try again.\n");
    }
}

return 0;
}

```

```
goku@goku1-ThinkPad-1460s:~/54/05/LabCycle/EXP9_Bankers Algorithm$ gcc banker.c
goku@goku1-ThinkPad-1460s:~/54/05/LabCycle/EXP9_Bankers Algorithm$ ./banker.out
```

Enter number of processes: 5

Enter number of resources: 3

Enter the Number of Total resources in the system:

for resource R0: 10

for resource R1: 5

for resource R2: 7

Enter the resources allocated to each process (allocated resource table):

0 1 0

2 0 0

3 0 2

2 1 1

0 0 2

Enter the Maximum resources needed by each process (Max claim table):

7 5 3

3 2 2

9 0 2

2 2 2

4 3

3

Menu:

1. Check Safe State (Banker's Algorithm)

2. Request Resources

3. Exit

Enter your choice: 1

System is in a safe state! Safe Sequence: P1 P3 P0 P2 P4

Menu:

1. Check Safe State (Banker's Algorithm)

2. Request Resources

3. Exit

Enter your choice: 2

Enter the process number (0 to 4): 1

Enter the resource request for process P1:

Resource R0: 1

Resource R1: 0

Resource R2: 2

System is in a safe state! Safe Sequence: P1 P3 P0 P2 P4

Resource request granted!

Menu:

1. Check Safe State (Banker's Algorithm)

2. Request Resources

3. Exit

Enter your choice: █

MEMORY ALLOCATION

BEST FIT

Program

```
#include <stdio.h>

typedef struct {
    int size;
    int isFree;
} MemoryBlock;

int bestFit(MemoryBlock blocks[], int numBlocks, int processSize) {
    int bestIndex = -1;
    int minSize = 999999;

    for (int i = 0; i < numBlocks; i++) {
        if (blocks[i].isFree && blocks[i].size >= processSize) {
            if (blocks[i].size < minSize) {
                minSize = blocks[i].size;
                bestIndex = i;
            }
        }
    }

    if (bestIndex != -1) {
        blocks[bestIndex].isFree = 0;
        printf("Process of size %d allocated in block of size %d\n", processSize,
blocks[bestIndex].size);
        return bestIndex;
    } else {
        printf("No suitable block found for process of size %d\n", processSize);
        return -1;
    }
}

int main() {
    MemoryBlock blocks[100];
    int numBlocks;

    printf("Enter the number of memory blocks: ");
    scanf("%d", &numBlocks);

    for (int i = 0; i < numBlocks; i++) {
        printf("Enter size of block %d: ", i + 1);
        scanf("%d", &blocks[i].size);
        blocks[i].isFree = 1;
    }

    int processSize;
    while (1) {
        printf("\nEnter the size of the process to allocate (0 to exit): ");
```

```
scanf("%d", &processSize);
if (processSize == 0) {
    break;
}
bestFit(blocks, numBlocks, processSize);
}

printf("\nRemaining memory blocks:\n");
for (int i = 0; i < numBlocks; i++) {
    printf("Block %d: Size = %d, Status = %s\n", i + 1, blocks[i].size, blocks[i].isFree ? "Free" :
"Allocated");
}

return 0;
}
```

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/Exp_11_MEMORYALLOCATION
ut
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ ./bestfit.out
Enter the number of memory blocks: 5
Enter size of block 1: 150
Enter size of block 2: 290
Enter size of block 3: 500
Enter size of block 4: 400
Enter size of block 5: 100

Enter the size of the process to allocate (0 to exit): 90
Process of size 90 allocated in block of size 100

Enter the size of the process to allocate (0 to exit): 180
Process of size 180 allocated in block of size 290

Enter the size of the process to allocate (0 to exit): 300
Process of size 300 allocated in block of size 400

Enter the size of the process to allocate (0 to exit): 480
Process of size 480 allocated in block of size 500

Enter the size of the process to allocate (0 to exit): 0

Remaining memory blocks:
Block 1: Size = 150, Status = Free
Block 2: Size = 290, Status = Allocated
Block 3: Size = 500, Status = Allocated
Block 4: Size = 400, Status = Allocated
Block 5: Size = 100, Status = Allocated
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$
```

WORSTFIT

program

```
#include <stdio.h>

typedef struct {
    int size;
    int isFree;
} MemoryBlock;

int worstfit(MemoryBlock blocks[], int numBlocks, int processSize) {
    int bestIndex = -1;
    int maxsize = -1;

    for (int i = 0; i < numBlocks; i++) {
        if (blocks[i].isFree && blocks[i].size >= processSize) {
            if (blocks[i].size > maxsize) {
                maxsize = blocks[i].size;
                bestIndex=i;
            }
        }
    }

    if (bestIndex != -1) {
        blocks[bestIndex].isFree = 0;
        printf("Process of size %d allocated in block of size %d\n", processSize,
blocks[bestIndex].size);
        return bestIndex;
    } else {
        printf("No suitable block found for process of size %d\n", processSize);
        return -1;
    }
}
```

```

int main() {
    MemoryBlock blocks[100];
    int numBlocks;

    printf("Enter the number of memory blocks: ");
    scanf("%d", &numBlocks);

    for (int i = 0; i < numBlocks; i++) {
        printf("Enter size of block %d: ", i + 1);
        scanf("%d", &blocks[i].size);
        blocks[i].isFree = 1;
    }
    int processSize;
    while (1) {
        printf("\nEnter the size of the process to allocate (0 to exit): ");
        scanf("%d", &processSize);
        if (processSize == 0) {
            break;
        }
        worstfit(blocks, numBlocks, processSize);
    }

    printf("\nRemaining memory blocks:\n");
    for (int i = 0; i < numBlocks; i++) {
        printf("Block %d: Size = %d, Status = %s\n", i + 1, blocks[i].size, blocks[i].isFree ? "Free" :
"Allocated");
    }

    return 0;
}

```

Output

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/Exp_11_MEMORYALLOCATION
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ gcc worstfit.c -o worstfit.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ ./worstfit.out
Enter the number of memory blocks: 5
Enter size of block 1: 150
Enter size of block 2: 290
Enter size of block 3: 500
Enter size of block 4: 400
Enter size of block 5: 100

Enter the size of the process to allocate (0 to exit): 90
Process of size 90 allocated in block of size 500

Enter the size of the process to allocate (0 to exit): 180
Process of size 180 allocated in block of size 400

Enter the size of the process to allocate (0 to exit): 300
No suitable block found for process of size 300

Enter the size of the process to allocate (0 to exit): 480
No suitable block found for process of size 480

Enter the size of the process to allocate (0 to exit): 0

Remaining memory blocks:
Block 1: Size = 150, Status = Free
Block 2: Size = 290, Status = Free
Block 3: Size = 500, Status = Allocated
Block 4: Size = 400, Status = Allocated
Block 5: Size = 100, Status = Free
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$
```

FIRST FIT

program

```
#include <stdio.h>

typedef struct {
    int size;
    int isFree;
} MemoryBlock;

int firstfit(MemoryBlock blocks[], int numBlocks, int processSize) {
    int bestIndex=-1,i;

    for (i=0;i<numBlocks;i++){
        if(blocks[i].isFree && blocks[i].size >= processSize){
            bestIndex = i;
            break;
        }
    }

    if (bestIndex != -1) {
        blocks[bestIndex].isFree = 0;
        printf("Process of size %d allocated in block of size %d\n", processSize,
blocks[bestIndex].size);
        return bestIndex;
    } else {
        printf("No suitable block found for process of size %d\n", processSize);
        return -1;
    }
}

int main() {
    MemoryBlock blocks[100];
    int numBlocks,index=0;

    printf("Enter the number of memory blocks: ");
    scanf("%d", &numBlocks);

    for (int i = 0; i < numBlocks; i++) {
        printf("Enter size of block %d: ", i + 1);
        scanf("%d", &blocks[i].size);
        blocks[i].isFree = 1;
    }

    int processSize;
    while (1) {
        printf("\nEnter the size of the process to allocate (0 to exit): ");
        scanf("%d", &processSize);
        if (processSize == 0) {
            break;
        }
        firstfit(blocks, numBlocks, processSize);
    }
}
```

```

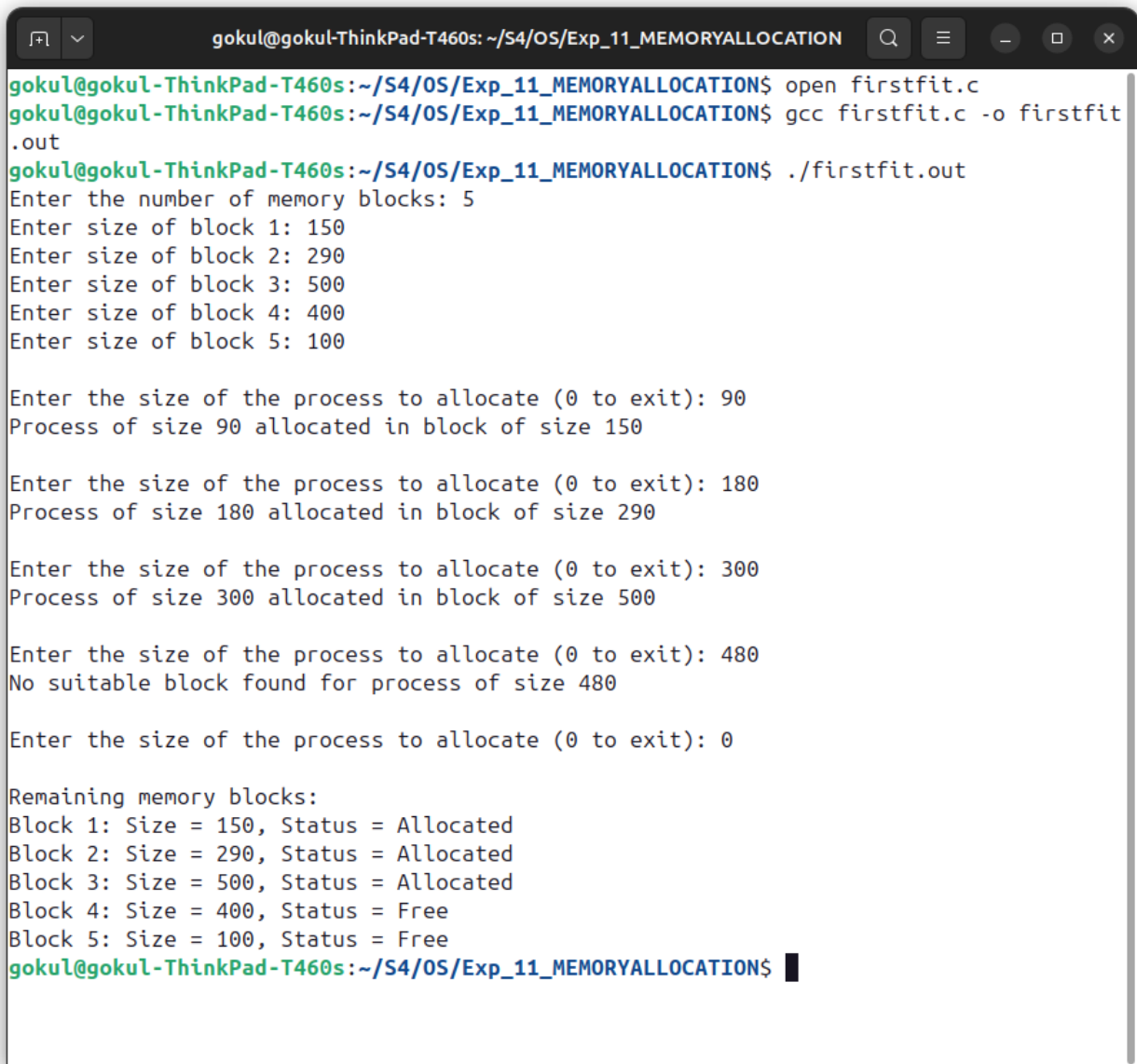
}

printf("\nRemaining memory blocks:\n");
for (int i = 0; i < numBlocks; i++) {
    printf("Block %d: Size = %d, Status = %s\n", i + 1, blocks[i].size, blocks[i].isFree ? "Free" :
"Allocated");
}

return 0;
}

```

Output



```

gokul@gokul-ThinkPad-T460s: ~/S4/OS/Exp_11_MEMORYALLOCATION
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ open firstfit.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ gcc firstfit.c -o firstfit.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$ ./firstfit.out
Enter the number of memory blocks: 5
Enter size of block 1: 150
Enter size of block 2: 290
Enter size of block 3: 500
Enter size of block 4: 400
Enter size of block 5: 100

Enter the size of the process to allocate (0 to exit): 90
Process of size 90 allocated in block of size 150

Enter the size of the process to allocate (0 to exit): 180
Process of size 180 allocated in block of size 290

Enter the size of the process to allocate (0 to exit): 300
Process of size 300 allocated in block of size 500

Enter the size of the process to allocate (0 to exit): 480
No suitable block found for process of size 480

Enter the size of the process to allocate (0 to exit): 0

Remaining memory blocks:
Block 1: Size = 150, Status = Allocated
Block 2: Size = 290, Status = Allocated
Block 3: Size = 500, Status = Allocated
Block 4: Size = 400, Status = Free
Block 5: Size = 100, Status = Free
gokul@gokul-ThinkPad-T460s:~/S4/OS/Exp_11_MEMORYALLOCATION$

```


FIRST IN FIRST OUT (FIFO)

Program:

```
#include <stdio.h>
```

```
void prfr(int frames[], int frame) {  
    int i;  
    printf("\tCurrent status of Frames: ");  
    for (i = 0; i < frame; i++) {  
        printf("%d ", frames[i]);  
    }  
    printf("\n");  
}
```

```
void fifo(int frame, int page[], int n) {  
    int i, j;  
    int count=0;  
    int frames[frame];  
    for (i = 0; i < frame; i++) {  
        frames[i] = -1;  
    }
```

```
    float pagefaults = 0, hits = 0;  
    int found;
```

```
    for (i = 0; i < n; i++) {  
        found = 0;  
        for (j = 0; j < frame; j++) {  
            if (page[i] == frames[j]) {  
                found = 1;  
                hits++;  
                printf("Page %d is already in the frames: ", page[i]);  
                prfr(frames, frame);  
                break;  
            }  
        }  
    }
```

```
    if (!found) {  
        pagefaults++;  
        frames[(count++) % frame] = page[i];  
        printf("Page %d loaded into frames:\t ", page[i]);  
        prfr(frames, frame);  
    }  
}
```

```

printf("Total Page Faults: %.0f\n", pagefaults);
printf("Total Hits: %.0f\n", hits);
printf("Miss Ratio: %.2f\n", pagefaults / n);
printf("Hit Ratio: %.2f\n", hits / n);
}

int main() {
    int i, n, frame, page[100];
    printf("Enter the Number of Frames: ");
    scanf("%d", &frame);

    printf("Enter the length of reference string: ");
    scanf("%d", &n);

    printf("Enter the reference string (space-separated): ");
    for (i = 0; i < n; i++) {
        scanf("%d", &page[i]);
    }

    fifo(frame, page, n);
    return 0;
}

```

Output:

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/LabCycle/EXP11_Pagerepla...
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ open fifo.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ gcc fifo.c
-o fifo.out
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ ./fifo.out

Enter the Number of Frames: 3
Enter the length of reference string: 10
Enter the reference string (space-separated): 1 2 3 3 2 1 4 7 8 1
Page 1 loaded into frames:           Current status of Frames: 1 -1 -1
Page 2 loaded into frames:           Current status of Frames: 1 2 -1
Page 3 loaded into frames:           Current status of Frames: 1 2 3
Page 3 is already in the frames:      Current status of Frames: 1 2 3
Page 2 is already in the frames:      Current status of Frames: 1 2 3
Page 1 is already in the frames:      Current status of Frames: 1 2 3
Page 4 loaded into frames:           Current status of Frames: 4 2 3
Page 7 loaded into frames:           Current status of Frames: 4 7 3
Page 8 loaded into frames:           Current status of Frames: 4 7 8
Page 1 loaded into frames:           Current status of Frames: 1 7 8
Total Page Faults: 7
Total Hits: 3
Miss Ratio: 0.70
Hit Ratio: 0.30
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$
```

Least Recently Used (LRU)

program

```
#include<stdio.h>

typedef struct{
    int word;
    int freq;
}frame;

void print(frame f[10],int n){
    int i;
    printf("\tCurrent status of frame [");
    for(i=0;i<n;i++){
        printf(" %d ",f[i].word);
    }
    printf("]\n");
}
```

```

void hit(frame f[10],int n,int i,int word){
    printf("HIT: The page %d is in the frame",word);
    //print(f,n);
    f[i].freq=n;
    for(i=0;i<n;i++){
        f[i].freq--;
    }
    print(f,n);
}

void miss(frame f[10],int n,int i,int word){
    int lowfreq=11,ind;
    printf("MISS: The page %d is not in the frame",word);
    for(i=0;i<n;i++){
        if(f[i].freq<lowfreq){
            ind=i;
            lowfreq=f[i].freq;
        }
    }
    f[ind].word=word;
    f[ind].freq=n;
    for(i=0;i<n;i++){
        f[i].freq--;
    }
    print(f,n);
}

```

```

void main(){
    int n,page,i,j;
    int pages[100];
    frame f[10];
    for(i=0;i<10;i++){
        f[i].word=-1;
        f[i].freq=i-10;
    }
    int flag;
    int hi=0,mi=0;
    printf("Enter the number of Frames in the system: ");
    scanf("%d",&n);
    printf("Enter the length of the referance string: ");
    scanf("%d", &page);
    printf("Enter the referance string (Space seperated): ");
    for(i=0;i<page;i++){
        scanf("%d",&pages[i]);
    }
    for(i=0;i<page;i++){

```

```

        flag=0;
        for(j=0;j<n;j++){
            if(f[j].word==pages[i]){
                hit(f,n,j,f[j].word);
                flag=1;
                hi++;
            }
        }
        if(flag==0){
            miss(f,n,i,pages[i]);
            mi++;
        }
    }
    printf("Number of Pagefaults: %d\n",mi);
    printf("Number of Hits: %d\n",hi);

    printf("Miss ratio: %d\n", (mi/page));
    printf("Hit ratio: %d\n", (hi/page));
    printf("Execution completed \n");
}

```

Output:

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/LabCycle/EXP11_Pagerepla...
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ open lru.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ gcc lru.c -o lru.out
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ ./lru.out
Enter the number of Frames in the system: 3
Enter the length of the reference string: 10
Enter the reference string (Space seperated): 1 2 3 3 2 1 4 7 8 1
MISS: The page 1 is not in the frame      Current status of frame [ 1  -1  -1 ]
MISS: The page 2 is not in the frame      Current status of frame [ 1  2  -1 ]
MISS: The page 3 is not in the frame      Current status of frame [ 1  2  3 ]
HIT: The page 3 is in the frame            Current status of frame [ 1  2  3 ]
HIT: The page 2 is in the frame            Current status of frame [ 1  2  3 ]
HIT: The page 1 is in the frame            Current status of frame [ 1  2  3 ]
MISS: The page 4 is not in the frame      Current status of frame [ 1  2  4 ]
MISS: The page 7 is not in the frame      Current status of frame [ 1  7  4 ]
MISS: The page 8 is not in the frame      Current status of frame [ 8  7  4 ]
MISS: The page 1 is not in the frame      Current status of frame [ 8  7  1 ]
Number of Pagefaults: 7
Number of Hits: 3
Miss ratio: 0
Hit ratio: 0
Execution completed
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$
```

Least Frequently Used (LFU)

program

```
#include<stdio.h>

typedef struct {
    int word;
    int freq;
} frame;

void print(frame f[10], int n) {
    int i;
    printf("\t Current status of frame [");
    for (i = 0; i < n; i++) {
        printf(" %d ", f[i].word);
    }
    printf("]\n");
}

void hit(frame f[10], int n, int i, int word) {
    printf("HIT: The page %d is in the frame ", word);
    f[i].freq++;
    print(f, n);
}

void miss(frame f[10], int n, int i, int word) {
    int lowfreq = 11, ind;
    printf("MISS: The page %d is not in the frame ", word);
    for (i = 0; i < n; i++) {
        if (f[i].freq < lowfreq) {
            ind = i;
            lowfreq = f[i].freq;
        }
    }
    f[ind].word = word;
    f[ind].freq = 1;
    print(f, n);
}
```

```

void main() {
    int n, page, i, j;
    int pages[100];
    frame f[10];
    for (i = 0; i < 10; i++) {
        f[i].word = -1;
        f[i].freq = 0;
    }
    int flag;
    int hi = 0, mi = 0;
    printf("Enter the number of Frames in the system: ");
    scanf("%d", &n);
    printf("Enter the length of the reference string: ");
    scanf("%d", &page);
    printf("Enter the reference string (Space separated): ");
    for (i = 0; i < page; i++) {
        scanf("%d", &pages[i]);
    }
    for (i = 0; i < page; i++) {
        flag = 0;
        for (j = 0; j < n; j++) {
            if (f[j].word == pages[i]) {
                hit(f, n, j, f[j].word);
                flag = 1;
                hi++;
            }
        }
        if (flag == 0) {
            miss(f, n, i, pages[i]);
            mi++;
        }
    }
    printf("Number of Pagefaults: %d\n", mi);
    printf("Number of Hits: %d\n", hi);

    printf("Miss ratio: %.2f\n", (float)(mi) / page);
    printf("Hit ratio: %.2f\n", (float)(hi) / page);
    printf("Execution completed \n");
}

```


Output:

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/LabCycle/EXP11_Pagerepla...
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ touch lfu.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ open lfu.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ gcc lfu.c -o lfu.out
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$ ./lfu.out
Enter the number of Frames in the system: 3
Enter the length of the reference string: 10
Enter the reference string (Space separated): 1 2 3 3 2 1 4 7 8 1
MISS: The page 1 is not in the frame      Current status of frame [ 1 -1 -1 ]
MISS: The page 2 is not in the frame      Current status of frame [ 1 2 -1 ]
MISS: The page 3 is not in the frame      Current status of frame [ 1 2 3 ]
HIT: The page 3 is in the frame           Current status of frame [ 1 2 3 ]
HIT: The page 2 is in the frame           Current status of frame [ 1 2 3 ]
HIT: The page 1 is in the frame           Current status of frame [ 1 2 3 ]
MISS: The page 4 is not in the frame      Current status of frame [ 4 2 3 ]
MISS: The page 7 is not in the frame      Current status of frame [ 7 2 3 ]
MISS: The page 8 is not in the frame      Current status of frame [ 8 2 3 ]
MISS: The page 1 is not in the frame      Current status of frame [ 1 2 3 ]
Number of Pagefaults: 7
Number of Hits: 3
Miss ratio: 0.70
Hit ratio: 0.30
Execution completed
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/LabCycle/EXP11_Pagereplacemnt$
```

First In First Out [FIFO]

Program

```
#include <stdio.h>
```

```
void prfr(int frames[], int frame) {  
    int i;  
    printf("\tCurrent status of Frames: ");  
    for (i = 0; i < frame; i++) {  
        printf("%d ", frames[i]);  
    }  
    printf("\n");  
}
```

```
void fifo(int frame, int page[], int n) {  
    int i, j;  
    int frames[frame];  
    for (i = 0; i < frame; i++) {  
        frames[i] = -1;  
    }  
  
    float pagefaults = 0, hits = 0;  
    int found;
```

```
    for (i = 0; i < n; i++) {  
        found = 0;  
        for (j = 0; j < frame; j++) {  
            if (page[i] == frames[j]) {  
                found = 1;  
                hits++;  
                printf("Page %d is already in the frames: ", page[i]);  
                prfr(frames, frame);  
                break;  
            }  
        }  
    }
```

```
    if (!found) {  
        pagefaults++;  
        frames[i % frame] = page[i];  
        printf("Page %d loaded into frames:\t ", page[i]);  
        prfr(frames, frame);  
    }  
}
```

```

printf("Total Page Faults: %.0f\n", pagefaults);
printf("Total Hits: %.0f\n", hits);
printf("Miss Ratio: %.2f\n", pagefaults / n);
printf("Hit Ratio: %.2f\n", hits / n);
}

int main() {
    int i, n, frame, page[100];
    printf("Enter the Number of Frames: ");
    scanf("%d", &frame);

    printf("Enter the length of reference string: ");
    scanf("%d", &n);

    printf("Enter the reference string (space-separated): ");
    for (i = 0; i < n; i++) {
        scanf("%d", &page[i]);
    }

    fifo(frame, page, n);
    return 0;
}

```

output

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP11_PageReplacement
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ open FIFO.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ gcc FIFO.c -o fifo.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ ./fifo.out
Enter the Number of Frames: 3
Enter the length of reference string: 20
Enter the reference string (space-separated): 1 2 3 2 3 1 8 3 1 3 1 3 9 8 7 6 1
2 3 4
Page 1 loaded into frames:           Current status of Frames: 1 -1 -1
Page 2 loaded into frames:           Current status of Frames: 1 2 -1
Page 3 loaded into frames:           Current status of Frames: 1 2 3
Page 2 is already in the frames:     Current status of Frames: 1 2 3
Page 3 is already in the frames:     Current status of Frames: 1 2 3
Page 1 is already in the frames:     Current status of Frames: 1 2 3
Page 8 loaded into frames:           Current status of Frames: 8 2 3
Page 3 is already in the frames:     Current status of Frames: 8 2 3
Page 1 loaded into frames:           Current status of Frames: 8 2 1
Page 3 loaded into frames:           Current status of Frames: 3 2 1
Page 1 is already in the frames:     Current status of Frames: 3 2 1
Page 3 is already in the frames:     Current status of Frames: 3 2 1
Page 9 loaded into frames:           Current status of Frames: 9 2 1
Page 8 loaded into frames:           Current status of Frames: 9 8 1
Page 7 loaded into frames:           Current status of Frames: 9 8 7
Page 6 loaded into frames:           Current status of Frames: 6 8 7
Page 1 loaded into frames:           Current status of Frames: 6 1 7
Page 2 loaded into frames:           Current status of Frames: 6 1 2
Page 3 loaded into frames:           Current status of Frames: 3 1 2
Page 4 loaded into frames:           Current status of Frames: 3 4 2
Total Page Faults: 14
Total Hits: 6
Miss Ratio: 0.70
Hit Ratio: 0.30
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$
```

Least Recently Used [LRU]

program

```
#include <stdio.h>
```

```
void prfr(int frames[], int frame) {
    int i;
    printf("\tCurrent status of Frames: ");
    for (i = 0; i < frame; i++) {
        printf("%d ", frames[i]);
    }
    printf("\n");
}
```

```
void lru(int frame, int page[], int n) {
    int i, j;
    int frames[frame];
    int last_used[frame];
    for (i = 0; i < frame; i++) {
        frames[i] = -1;
        last_used[i] = -1;
    }
```

```
float pagefaults = 0, hits = 0;
int found, lru_index;
```

```
for (i = 0; i < n; i++) {
    found = 0;
    for (j = 0; j < frame; j++) {
        if (page[i] == frames[j]) {
            found = 1;
            hits++;
            last_used[j] = i;
            printf("Page %d is already in the frames: ", page[i]);
            prfr(frames, frame);
            break;
        }
    }
}
```

```
if (!found) {
    pagefaults++;
    lru_index = 0;
    for (j = 1; j < frame; j++) {
        if (last_used[j] < last_used[lru_index]) {
            lru_index = j;
        }
    }
}
```

```

        frames[lru_index] = page[i];
        last_used[lru_index] = i;
        printf("Page %d loaded into frames: ", page[i]);
        prfr(frames, frame);
    }
}

printf("Total Page Faults: %.0f\n", pagefaults);
printf("Total Hits: %.0f\n", hits);
printf("Miss Ratio: %.2f\n", pagefaults / n);
printf("Hit Ratio: %.2f\n", hits / n);
}

int main() {
    int i, n, frame, page[100];
    printf("Enter the Number of Frames: ");
    scanf("%d", &frame);

    printf("Enter the length of reference string: ");
    scanf("%d", &n);

    printf("Enter the reference string (space-separated): ");
    for (i = 0; i < n; i++) {
        scanf("%d", &page[i]);
    }

    lru(frame, page, n);
    return 0;
}

```

Output

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP11_PageReplacement
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ open LRU.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ gcc LRU.c -o lru.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ ./lru.out
Enter the Number of Frames: 3
Enter the length of reference string: 20
Enter the reference string (space-separated): 1 2 3 2 3 1 8 3 1 3 1 3 9 8 7 6 1
2 3 4
Page 1 loaded into frames:      Current status of Frames: 1 -1 -1
Page 2 loaded into frames:      Current status of Frames: 1 2 -1
Page 3 loaded into frames:      Current status of Frames: 1 2 3
Page 2 is already in the frames: Current status of Frames: 1 2 3
Page 3 is already in the frames: Current status of Frames: 1 2 3
Page 1 is already in the frames: Current status of Frames: 1 2 3
Page 8 loaded into frames:      Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 1 is already in the frames: Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 1 is already in the frames: Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 9 loaded into frames:      Current status of Frames: 1 9 3
Page 8 loaded into frames:      Current status of Frames: 8 9 3
Page 7 loaded into frames:      Current status of Frames: 8 9 7
Page 6 loaded into frames:      Current status of Frames: 8 6 7
Page 1 loaded into frames:      Current status of Frames: 1 6 7
Page 2 loaded into frames:      Current status of Frames: 1 6 2
Page 3 loaded into frames:      Current status of Frames: 1 3 2
Page 4 loaded into frames:      Current status of Frames: 4 3 2
Total Page Faults: 12
Total Hits: 8
Miss Ratio: 0.60
Hit Ratio: 0.40
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$
```

Least Frequently Used[LFU]

Program

```
#include <stdio.h>
```

```
void prfr(int frames[], int frame) {  
    int i;  
    printf("\tCurrent status of Frames: ");  
    for (i = 0; i < frame; i++) {  
        printf("%d ", frames[i]);  
    }  
    printf("\n");  
}
```

```
void lfu(int frame, int page[], int n) {  
    int i, j;  
    int frames[frame];  
    int frequency[frame];  
    int last_used[frame];
```

```
    for (i = 0; i < frame; i++) {  
        frames[i] = -1;  
        frequency[i] = 0;  
        last_used[i] = -1;  
    }
```

```
    float pagefaults = 0, hits = 0;  
    int found, lfu_index;
```

```
    for (i = 0; i < n; i++) {  
        found = 0;  
        for (j = 0; j < frame; j++) {  
            if (page[i] == frames[j]) {  
                found = 1;  
                hits++;  
                frequency[j]++;  
                last_used[j] = i;  
                printf("Page %d is already in the frames: ", page[i]);  
                prfr(frames, frame);  
                break;  
            }  
        }  
    }
```



```

        if (!found) {
            pagefaults++;
            lfu_index = 0;
            for (j = 1; j < frame; j++) {
                if (frequency[j] < frequency[lfu_index] || (frequency[j] == frequency[lfu_index] &&
last_used[j] < last_used[lfu_index])) {
                    lfu_index = j;
                }
            }
            frames[lfu_index] = page[i];
            frequency[lfu_index] = 1;
            last_used[lfu_index] = i;
            printf("Page %d loaded into frames: ", page[i]);
            prfr(frames, frame);
        }
    }

    printf("Total Page Faults: %.0f\n", pagefaults);
    printf("Total Hits: %.0f\n", hits);
    printf("Miss Ratio: %.2f\n", pagefaults / n);
    printf("Hit Ratio: %.2f\n", hits / n);
}

int main() {
    int i, n, frame, page[100];
    printf("Enter the Number of Frames: ");
    scanf("%d", &frame);

    printf("Enter the length of reference string: ");
    scanf("%d", &n);

    printf("Enter the reference string (space-separated): ");
    for (i = 0; i < n; i++) {
        scanf("%d", &page[i]);
    }

    lfu(frame, page, n);
    return 0;
}

```

Output

```
gokul@gokul-ThinkPad-T460s: ~/S4/OS/EXP11_PageReplacement
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ open LFU.c
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ gcc LFU.c -o LFU.out
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$ ./LFU.out
Enter the Number of Frames: 3
Enter the length of reference string: 20
Enter the reference string (space-separated): 1 2 3 2 3 1 8 3 1 3 1 3 9 8 7 6 1
2 3 4
Page 1 loaded into frames:      Current status of Frames: 1 -1 -1
Page 2 loaded into frames:      Current status of Frames: 1 2 -1
Page 3 loaded into frames:      Current status of Frames: 1 2 3
Page 2 is already in the frames: Current status of Frames: 1 2 3
Page 3 is already in the frames: Current status of Frames: 1 2 3
Page 1 is already in the frames: Current status of Frames: 1 2 3
Page 8 loaded into frames:      Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 1 is already in the frames: Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 1 is already in the frames: Current status of Frames: 1 8 3
Page 3 is already in the frames: Current status of Frames: 1 8 3
Page 9 loaded into frames:      Current status of Frames: 1 9 3
Page 8 loaded into frames:      Current status of Frames: 1 8 3
Page 7 loaded into frames:      Current status of Frames: 1 7 3
Page 6 loaded into frames:      Current status of Frames: 1 6 3
Page 1 is already in the frames: Current status of Frames: 1 6 3
Page 2 loaded into frames:      Current status of Frames: 1 2 3
Page 3 is already in the frames: Current status of Frames: 1 2 3
Page 4 loaded into frames:      Current status of Frames: 1 4 3
Total Page Faults: 10
Total Hits: 10
Miss Ratio: 0.50
Hit Ratio: 0.50
gokul@gokul-ThinkPad-T460s:~/S4/OS/EXP11_PageReplacement$
```

First Come First Serve (FCFS)

Program

```
#include <stdio.h>
#include <stdlib.h>

void printSeekSequence(int sequence[], int n, int init) {
    int i;
    printf("Seek Sequence: ");
    printf("%d -> ",init);
    for (i = 0; i < n - 1; i++) {
        printf("%d -> ", sequence[i]);
    }
    printf("%d\n", sequence[n - 1]);
}

void fcfs(int tracks[], int n, int initial, int totalCylinders) {
    int totalSeekDistance = 0, i;
    int seekSequence[n];
    int current = initial;
    seekSequence[0] = current;

    for (i = 0; i < n; i++) {
        if (tracks[i] < 0 || tracks[i] >= totalCylinders) {
            printf("Error: Disk request %d is out of bounds (valid range: 0 to %d).\n", tracks[i],
totalCylinders - 1);
            return;
        }

        for (i = 0; i < n; i++) {
            totalSeekDistance += abs(current - tracks[i]);
            current = tracks[i];
            seekSequence[i] = current;
        }

        float avgSeekDistance = (float)totalSeekDistance / n;

        printf("Total Seek Distance: %d\n", totalSeekDistance);
        printf("Average Seek Distance: %.2f\n", avgSeekDistance);
        printSeekSequence(seekSequence, n,initial);
    }
}
```

```

int main() {
    int n, initial, totalCylinders, i;

    printf("Enter the number of cylinders: ");
    scanf("%d", &totalCylinders);

    printf("Enter the number of disk requests: ");
    scanf("%d", &n);

    int tracks[n];

    printf("Enter the disk requests (space-separated): ");
    for (i = 0; i < n; i++) {
        scanf("%d", &tracks[i]);
    }

    printf("Enter the initial head position: ");
    scanf("%d", &initial);

    if (initial < 0 || initial >= totalCylinders) {
        printf("Error: Initial head position is out of bounds (valid range: 0 to %d).\n", totalCylinders -
1);
        return 1;
    }

    fcfs(tracks, n, initial, totalCylinders);
    return 0;
}

```

Output

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/EXP12_Disk sched...  
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ open fcfs.c  
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ gcc fcfs.c -o  
fcfs.out  
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ ./fcfs.out  
Enter the number of cylinders: 200  
Enter the number of disk requests: 8  
Enter the disk requests (space-separated): 98 183 37 122 14 124 65 67  
Enter the initial head position: 53  
Total Seek Distance: 640  
Average Seek Distance: 80.00  
Seek Sequence: 53 -> 98 -> 183 -> 37 -> 122 -> 14 -> 124 -> 65 -> 67  
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$
```

SCAN (ELEVATOR)

Program

```
#include <stdio.h>
#include <stdlib.h>

void printSeekSequence(int sequence[], int n, int init) {
    printf("Seek Sequence: %d", init);
    for (int i = 0; i < n; i++) {
        printf(" -> %d", sequence[i]);
    }
    printf("\n");
}

void scan(int tracks[], int n, int initial, int totalCylinders, char direction) {
    int totalSeekDistance = 0;
    int seekSequence[n + 2];
    int current = initial;
    int count = 0;

    for (int i = 0; i < n; i++) {
        if (tracks[i] < 0 || tracks[i] >= totalCylinders) {
            printf("Error: Disk request %d is out of bounds (valid range: 0 to %d).\n", tracks[i],
totalCylinders - 1);
            return;
        }
    }

    // Sorting the request array
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (tracks[i] > tracks[j]) {
                int temp = tracks[i];
                tracks[i] = tracks[j];
                tracks[j] = temp;
            }
        }
    }

    int left[n], right[n], leftCount = 0, rightCount = 0;

    for (int i = 0; i < n; i++) {
        if (tracks[i] < initial) {
            left[leftCount++] = tracks[i];
        } else {
            right[rightCount++] = tracks[i];
        }
    }

    int seek = 0;
    if (direction == 'R' || direction == 'r') {
```

```

    for (int i = 0; i < rightCount; i++) {
        seekSequence[count++] = right[i];
        seek += abs(current - right[i]);
        current = right[i];
    }
    if (rightCount > 0 && leftCount > 0) {
        seekSequence[count++] = totalCylinders - 1;
        seek += abs(current - (totalCylinders - 1));
        current = totalCylinders - 1;
    }
    for (int i = leftCount - 1; i >= 0; i--) {
        seekSequence[count++] = left[i];
        seek += abs(current - left[i]);
        current = left[i];
    }
}
else if (direction == 'L' || direction == 'l') {
    for (int i = leftCount - 1; i >= 0; i--) {
        seekSequence[count++] = left[i];
        seek += abs(current - left[i]);
        current = left[i];
    }
    if (leftCount > 0 && rightCount > 0) {
        seekSequence[count++] = 0;
        seek += abs(current - 0);
        current = 0;
    }
    for (int i = 0; i < rightCount; i++) {
        seekSequence[count++] = right[i];
        seek += abs(current - right[i]);
        current = right[i];
    }
} else {
    printf("Error: Invalid direction input. Please enter 'L' for left or 'R' for right.\n");
    return;
}

printf("Total Seek Distance: %d\n", seek);
printf("Average Seek Distance: %.2f\n", (float)seek / n);
printSeekSequence(seekSequence, count, initial);
}

```

```

int main() {
    int n, initial, totalCylinders;
    char direction;

    printf("Enter the number of cylinders: ");
    scanf("%d", &totalCylinders);

    printf("Enter the number of disk requests: ");
    scanf("%d", &n);

    int tracks[n];
    printf("Enter the disk requests (space-separated): ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &tracks[i]);
    }

    printf("Enter the initial head position: ");
    scanf("%d", &initial);
    if (initial < 0 || initial >= totalCylinders) {
        printf("Error: Initial head position is out of bounds (valid range: 0 to %d).\n", totalCylinders -
1);
        return 1;
    }

    printf("Enter the initial direction (L for left, R for right): ");
    scanf(" %c", &direction);

    scan(tracks, n, initial, totalCylinders, direction);
    return 0;
}

```


Output

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/EXP12_Disk sched...
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ open scan.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ gcc scan.c -o
scan.out
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ ./scan.out
Enter the number of cylinders: 200
Enter the number of disk requests: 8
Enter the disk requests (space-separated): 98 183 37 122 14 124 65 67
Enter the initial head position: 53
Enter the initial direction (L for left, R for right): L
Total Seek Distance: 236
Average Seek Distance: 29.50
Seek Sequence: 53 -> 37 -> 14 -> 0 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ ./scan.out
Enter the number of cylinders: 200
Enter the number of disk requests: 8
Enter the disk requests (space-separated): 98 183 37 122 14 124 65 67
Enter the initial head position: 53
Enter the initial direction (L for left, R for right): R
Total Seek Distance: 331
Average Seek Distance: 41.38
Seek Sequence: 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 199 -> 37 -> 14
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$
```

Circular SCAN (Circular Elevator)

Program

```
#include <stdio.h>
#include <stdlib.h>

void printSeekSequence(int sequence[], int n, int init) {
    printf("Seek Sequence: %d", init);
    for (int i = 0; i < n; i++) {
        printf(" -> %d", sequence[i]);
    }
    printf("\n");
}

void cscan(int tracks[], int n, int initial, int totalCylinders, char direction) {
    int totalSeekDistance = 0;
    int seekSequence[n + 2];
    int current = initial;
    int count = 0;

    for (int i = 0; i < n; i++) {
        if (tracks[i] < 0 || tracks[i] >= totalCylinders) {
            printf("Error: Disk request %d is out of bounds (valid range: 0 to %d).\n", tracks[i],
totalCylinders - 1);
            return;
        }
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (tracks[i] > tracks[j]) {
                int temp = tracks[i];
                tracks[i] = tracks[j];
                tracks[j] = temp;
            }
        }
    }

    int left[n], right[n], leftCount = 0, rightCount = 0;
```

```

for (int i = 0; i < n; i++) {
    if (tracks[i] < initial) {
        left[leftCount++] = tracks[i];
    } else {
        right[rightCount++] = tracks[i];
    }
}

int seek = 0;
if (direction == 'R' || direction == 'r') {
    for (int i = 0; i < rightCount; i++) {
        seekSequence[count++] = right[i];
        seek += abs(current - right[i]);
        current = right[i];
    }
    if (rightCount > 0) {
        seekSequence[count++] = totalCylinders - 1;
        seek += abs(current - (totalCylinders - 1));
        current = 0;
        seekSequence[count++] = 0;
        seek += totalCylinders - 1; // Add the jump distance
    }
    for (int i = 0; i < leftCount; i++) {
        seekSequence[count++] = left[i];
        seek += abs(current - left[i]);
        current = left[i];
    }
}
else if (direction == 'L' || direction == 'l') {
    for (int i = leftCount - 1; i >= 0; i--) {
        seekSequence[count++] = left[i];
        seek += abs(current - left[i]);
        current = left[i];
    }
    if (leftCount > 0) {
        seekSequence[count++] = 0;
        seek += abs(current - 0);
        current = totalCylinders - 1;
        seekSequence[count++] = totalCylinders - 1;
        seek += totalCylinders - 1; // Add the jump distance
    }
}

```

```

        for (int i = rightCount - 1; i >= 0; i--) {
            seekSequence[count++] = right[i];
            seek += abs(current - right[i]);
            current = right[i];
        }
    } else {
        printf("Error: Invalid direction input. Please enter 'L' for left or 'R' for right.\n");
        return;
    }

    printf("Total Seek Distance: %d\n", seek);
    printf("Average Seek Distance: %.2f\n", (float)seek / n);
    printSeekSequence(seekSequence, count, initial);
}

int main() {
    int n, initial, totalCylinders;
    char direction;

    printf("Enter the number of cylinders: ");
    scanf("%d", &totalCylinders);

    printf("Enter the number of disk requests: ");
    scanf("%d", &n);

    int tracks[n];
    printf("Enter the disk requests (space-separated): ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &tracks[i]);
    }

    printf("Enter the initial head position: ");
    scanf("%d", &initial);
    if (initial < 0 || initial >= totalCylinders) {
        printf("Error: Initial head position is out of bounds (valid range: 0 to %d).\n", totalCylinders -
1);
        return 1;
    }

    printf("Enter the initial direction (L for left, R for right): ");
    scanf(" %c", &direction);

    cscan(tracks, n, initial, totalCylinders, direction);
    return 0;
}

```

OUTPUT

```
gokulp@gokulp-B365M-GAMING-HD: ~/ASIET-main/S4/OS/EXP12_Disk sched...
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ open cscan.c
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ gcc cscan.c -o cscan.out
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ ./cscan.out
Enter the number of cylinders: 200
Enter the number of disk requests: 8
Enter the disk requests (space-separated): 98 183 37 122 14 124 65 67
Enter the initial head position: 53
Enter the initial direction (L for left, R for right): L
Total Seek Distance: 386
Average Seek Distance: 48.25
Seek Sequence: 53 -> 37 -> 14 -> 0 -> 199 -> 183 -> 124 -> 122 -> 98 -> 67 -> 65
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$ ./cscan.out
Enter the number of cylinders: 200
Enter the number of disk requests: 8
Enter the disk requests (space-separated): 98 183 37 122 14 124 65 67
Enter the initial head position: 53
Enter the initial direction (L for left, R for right): R
Total Seek Distance: 382
Average Seek Distance: 47.75
Seek Sequence: 53 -> 65 -> 67 -> 98 -> 122 -> 124 -> 183 -> 199 -> 0 -> 14 -> 37
gokulp@gokulp-B365M-GAMING-HD:~/ASIET-main/S4/OS/EXP12_Disk scheduling$
```