SCHOOL OF ENGINEERING

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI – 682022, KERALA

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

19-202-0610 OPERATING SYSTEM LABORATORY

Name: MEENAKSHI M KUMAR
Register No.: 20222066
Batch: CS B

Academic Year: 2025

INDEX

EXP 1(C): IMPLEMENTATION OF LINUX COMMANDS

AIM: To Execute linux commands to perform:

(a) Create a folder COMPUTER_SCIENCE_AND_ENGINEERING(CSE).
(b) Create two sub folders S3 and S4 inside CSE folder.
(c) Inside S3 folder, create 3 files teacher and student using touch command.
(d) Add name, roll no: and mark in student file.
(e) In teacher file, add name.
(f) View both files using cat command.
(g) Add a new value department to teacher file using cat command.
(h) Copy the data in teacher file to student file.
(i) Search any word “Anu” in student file.
(j) Display last 10 lines from student file.
(k) List all the files.
(l) Change teacher filename to staff.
(m) Delete all the files.
(n) Delete all the folders.

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop$ mkdir MEENAKSHI_B2_62
cs20222066@NOSLAB108:~/Desktop$ cd MEENAKSHI_B2_62
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ mkdir COMPUTER_SCIENCE_AND_ENGINEERING
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ cd COMPUTER_SCIENCE_AND_ENGINEERING
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING$cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMP
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING$ mkdir S3
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING$ mkdir S4
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING$ cd S3
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ touch teacher.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ touch student.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat student.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat > student.txt
NAME            ROLL_NO:                MARK
Aaliya            01                     45
Bison             02                     47
Anu               03                     36
Vishaka           04                     42
Salman            05                     39
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat student.txt
NAME            ROLL_NO:                MARK
Aaliya            01                     45
Bison             02                     47
Anu               03                     36
Vishaka           04                     42
Salman            05                     39
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat > teacher.txt
NAME
Anupama
Rajesh
Manu
Lena
Shekar
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat teacher.txt
NAME
Anupama
Rajesh
Manu
Lena
Shekar
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat student.txt
NAME            ROLL_NO:                MARK
Aaliya            01                     45
Bison             02                     47
Anu               03                     36
Vishaka           04                     42
Salman            05                     39
NAME
Anupama
Rajesh
Manu
Lena
Shekar

DEPARTMENT
Anupama - CS
Rajesh - EC
Manu - ME
Lena - CS
Shekar - CE
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ grep "Anu" student.txt
Anu               03                     36
Anupama
Anupama - CS
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ tail student.txt
Manu
Lena
Shekar

DEPARTMENT
Anupama - CS
Rajesh - EC
Manu - ME
Lena - CS
Shekar - CE
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ ls -l
total 8
-rw-r--r-- 1 cs20222066 CS2022 219 Jan 20 14:21 student.txt
-rw-r--r-- 1 cs20222066 CS2022 106 Jan 20 14:09 teacher.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ mv teacher.txt staff.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ rm staff.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ rm student.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat >> teacher.txt
DEPARTMENT
Anupama - CS
Rajesh - EC
Manu - ME
Lena - CS
Shekar - CE
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cp -n teacher.txt student.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat student.txt
NAME            ROLL_NO:                MARK
Aaliya            01                     45
Bison             02                     47
Anu               03                     36
Vishaka           04                     42
Salman            05                     39
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat teacher.txt
NAME
Anupama
Rajesh
Manu
Lena
Shekar

DEPARTMENT
Anupama - CS
Rajesh - EC
Manu - ME
Lena - CS
Shekar - CE
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat teacher.txt >> student.txt
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cat teacher.txt
NAME
Anupama
Rajesh
Manu
Lena
Shekar

DEPARTMENT
Anupama - CS
Rajesh - EC
Manu - ME
Lena - CS
Shekar - CE
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62/COMPUTER_SCIENCE_AND_ENGINEERING/S3$ cd
cs20222066@NOSLAB108:~$ rm -d S3
rm: cannot remove 'S3': No such file or directory
cs20222066@NOSLAB108:~$ ls -l
total 36
drwxr-xr-x 4 cs20222066 CS2022 4096 Jan 20 13:24 Desktop
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Documents
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Downloads
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Music
drwxr-xr-x 3 cs20222066 CS2022 4096 Oct 29 15:31 Pictures
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Public
drwx------ 4 cs20222066 CS2022 4096 Oct 29 15:56 snap
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Templates
drwxr-xr-x 2 cs20222066 CS2022 4096 Oct 22 13:10 Videos
cs20222066@NOSLAB108:~$ cd Desktop/MEENAKSHI_B2_62
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ls -l
total 4
drwxr-xr-x 4 cs20222066 CS2022 4096 Jan 20 13:26 COMPUTER_SCIENCE_AND_ENGINEERING
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ rm -d COMPUTER_SCIENCE_AND_ENIGINEERING
rm: cannot remove 'COMPUTER_SCIENCE_AND_ENIGINEERING': No such file or directory
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ rm -d *
rm: cannot remove 'COMPUTER_SCIENCE_AND_ENGINEERING': Directory not empty
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ rm -r *
```

EXP 3: SHELL PROGRAMS

AIM: To write programs in shell script to do the following:

(a) Perform basic arithmetic operations:

```
echo "Enter 2 numbers(space-separated):"
read a b
zero=0
echo "Enter operator(1:+,2:-,3:*,4:/):"
read op
case $op in
        1)echo "Sum:$((a+b))";;
        2)echo "Difference:$((a-b))";;
        3)echo "Product:$((a*b))";;
        4)if [ $b -ne $zero ]; then
                echo "Quotient:$((a/b))"
          else
                echo "Division not possible!"
          fi
esac
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit arithmetic.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x arithmetic.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./arithmetic.sh
Enter 2 numbers(space-separated):
5 6
Enter operator(1:+,2:-,3:*,4:/):
1
Sum:11
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./arithmetic.sh
Enter 2 numbers(space-separated):
5 6
Enter operator(1:+,2:-,3:*,4:/):
2
Difference:-1
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./arithmetic.sh
Enter 2 numbers(space-separated):
5 6
Enter operator(1:+,2:-,3:*,4:/):
3
Product:30
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./arithmetic.sh
Enter 2 numbers(space-separated):
6 5
Enter operator(1:+,2:-,3:*,4:/):
4
Quotient:1
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./arithmetic.sh
Enter 2 numbers(space-separated):
6 0
Enter operator(1:+,2:-,3:*,4:/):
4
Division not possible!
```

(b) Find Largest of 3 numbers

```
echo "Enter 3 numbers(space-separated):"
read a b c
if [ $a -ge $b ] && [ $a -ge $c ]; then
        echo "$a is the largest"
elif [ $b -ge $c ]; then
        echo "$b is the largest"
else
        echo "$c is the largest"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit large3.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x large3.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./large3.sh
Enter 3 numbers(space-separated):
65 8 9
65 is the largest
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./large3.sh
Enter 3 numbers(space-separated):
8 65 7
65 is the largest
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./large3.sh
Enter 3 numbers(space-separated):
45 32 61
61 is the largest
```

(c) Swap Two Numbers

```
echo "Enter 2 numbers(space-separated):"
read a b
echo "Before swapping:a=$a, b=$b"
temp=$a
a=$b
b=$temp
echo "After swapping:a=$a, b=$b"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit swap.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x swap.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./swap.sh
Enter 2 numbers(space-separated):
45 36
Before swapping:a=45, b=36
After swapping:a=36, b=45
```

(d) Check Even/Odd

```
echo "Enter a number:"
read num
if [ $((num%2)) -eq 0 ]; then
        echo "$num is Even!"
else
        echo "$num is Odd!"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit evenodd.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x evenodd.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./evenodd.sh
Enter a number:
21
21 is Odd!
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./evenodd.sh
Enter a number:
36
36 is Even!
```

(e) Find factorial of a Number

```
echo "Enter a number:"
read num
temp=$num
fact=1
while [ $num -gt 0 ]; do
        fact=$((fact*num))
        num=$((num-1))
done
echo "Factorial of $temp :$fact"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit factorial.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x factorial.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./factorial.sh
Enter a number:
5
Factorial of 5 :120
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./factorial.sh
Enter a number:
0
Factorial of 0 :1
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./factorial.sh
Enter a number:
6
Factorial of 6 :720
```

(f) Check Prime/Not

```
echo "Enter a number:"
read num
is_prime=1
for ((i=2 ; i*i<=num ; i++)); do
	if [ $((num%i)) -eq 0 ]; then
		is_prime=0
		break
	fi
done
if [ $num -lt 2 ]; then
	is_prime=0
fi
if [ $is_prime -eq 1 ]; then
	echo "$num is Prime"
else
	echo "$num is Not Prime"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit prime.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x prime.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./prime.sh
Enter a number:
13
13 is Prime
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./prime.sh
Enter a number:
6
6 is Not Prime
```

(g) Find Sum of Digits

```
echo "Enter a number:"
read num
temp=$num
sum=0
while [ $num -gt 0 ]; do
	digit=$((num%10))
	sum=$((sum+digit))
	num=$((num/10))
done
echo "Sum of digits of $temp:$sum"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit sum.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x sum.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./sum.sh
Enter a number:
256
Sum of digits of 256:13
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./sum.sh
Enter a number:
65
Sum of digits of 65:11
```

(h) Display Fibonacci Series

```
echo "Enter the number of terms:"
read n
a=0
b=1
echo "Fibonacci Series of $n terms:"
for ((i=0;i<n;i++)); do
        echo -n "$a "
        next=$((a+b))
        a=$b
        b=$next
done
echo
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit fibonacci.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x fibonacci.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./fibonacci.sh
Enter the number of terms:
10
Fibonacci Series of 10 terms:
0 1 1 2 3 5 8 13 21 34
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./fibonacci.sh
Enter the number of terms:
5
Fibonacci Series of 5 terms:
0 1 1 2 3
```

(i) Check Palindrome/Not in a Number

```
echo "Enter a number:"
read num
ori=$num
rev=0
while [ $num -gt 0 ]; do
        digit=$((num%10))
        rev=$((rev*10+digit))
        num=$((num/10))
done
if [ $rev -eq $ori ]; then
        echo "$ori is a Palindrome!"
else
        echo "$ori is Not a Palindrome!"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit palindnum.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x palindnum.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./palindnum.sh
Enter a number:
151
151 is a Palindrome!
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./palindnum.sh
Enter a number:
253
253 is Not a Palindrome!
```

(j) Sort a Given Array of Numbers:

```
echo "Enter the array elements(space-separated):"
read -a arr
n=${#arr[@]}
for ((i=0;i<n;i++)); do
        for ((j=0;j<n-i-1;j++)); do
                if [ ${arr[j]} -gt ${arr[j+1]} ]; then
                        temp=${arr[j]}
                        arr[j]=${arr[j+1]}
                        arr[j+1]=$temp
                fi
        done
done
echo "Sorted array: ${arr[@]}"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit sortarray.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x sortarray.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./sortarray.sh
Enter the array elements(space-separated):
45 2 15 68 32 45 78 12
Sorted array: 2 12 15 32 45 45 68 78
```

(k) Reverse a string

```
echo "Enter a string:"
read str
echo "Reversed String:$(echo $str|rev)"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit revstring.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x revstring.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./revstring.sh
Enter a string:
Lion
Reversed String:noiL
```

(l) Concatenate Two Strings

```
echo "Enter first string:"
read str1
echo "Enter second string:"
read str2
concat="$str1$str2"
echo "Concatenated string:$concat"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit concatstr.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x concatstr.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./concatstr.sh
Enter first string:
Computer
Enter second string:
Science
Concatenated string:ComputerScience
```

(m) Count occurence of a character in a string

```
echo "Enter a string:"
read str
echo "Enter a character to count:"
read char
count=$(echo -n "$str"|grep -o "$char"|wc -l)
echo "Occurences of '$char':$count"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit substrpos.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./charoccur.sh
Enter a string:
malAyalam
Enter a character to count:
a
Occurences of 'a':4
```

(n) Count occurence of a word in a file

```
echo "Enter the word to count:"
read word
echo "Enter the filename:"
read file
count=$(grep -iow $word $file |wc -l)
echo "Occurance of '$word' in $file:$count"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./wordoccurfile.sh
Enter the word to count:
App
Enter the filename:
wordoccurfile.txt
Occurance of 'App' in wordoccurfile.txt:2
```

(o) Check if two strings are equal

```
echo "Enter first string:"
read str1
echo "Enter second string:"
read str2
if [ "$str1" = "$str2" ]; then
        echo "Strings are equal"
```

```
else
        echo "Strings are not equal"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit strequal.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./strequal.sh
Enter first string:
Scool
Enter second string:
School
Strings are not equal
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./strequal.sh
Enter first string:
School
Enter second string:
School
Strings are equal
```

(p)Convert a string to uppercase

```
echo "Enter a string:"
read str
upper="${str^^}"
echo "Uppercase:"$upper
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit strupper.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./strupper.sh
Enter a string:
Melisa
Uppercase:MELISA
```

(q) Replace a substring

```
echo "Enter original string:"
read str
echo "Enter substring to replace:"
read old
echo "Enter the new substring:"
read new
echo "Modified string:$(echo $str | sed "s/$old/$new/g")"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./replstr.sh
Enter original string:
computer science
Enter substring to replace:
science
Enter the new substring:
engineer
Modified string:computer engineer
```

(r) Check if a string is palindrome

```
echo "Enter  a string:"
read str
rev_str=$(echo $str|rev)
if [ "$str" = "$rev_str" ]; then
        echo "Palindrome"
else
        echo "Not Palindrome"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit palindstr.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x palindstr.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./palindstr.sh
Enter  a string:
eve
Palindrome
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./palindstr.sh
Enter  a string:
liya
Not Palindrome
```

(s) Remove leading and trailing white spaces

```
echo "Enter string with spaces:"
read str
trimmed=$(echo "$str"|sed 's/^ *//; s/ *$//')
echo "Trimmed string:'$trimmed'"
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit remleadtrailsp.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x remleadtrailsp.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./remleadtrailsp.sh
Enter string with spaces:
     Melisa
Trimmed string:'Melisa'
```

(t) Find the position of a substring in a string

```
echo "Enter main string:"
read main
echo "Enter substring to find:"
read sub
pos=$(expr index "$main" "$sub")
if [ "$pos" -eq 0 ]; then
        echo "Substring not found"
else
        echo "Substring found at position:$pos"
fi
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit substrpos.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ chmod +x substrpos.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./substrpos.sh
Enter main string:
Computer
Enter substring to find:
mpu
Substring found at position:3
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./substrpos.sh
Enter main string:
Computer
Enter substring to find:
all
Substring not found
```

EXP 4: PROCESS CREATION USING fork() SYSTEM CALL

AIM: To familiarize the fork() and getpid() system calls.

PROGRAM

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
int main()
{
 int p=fork();
 if (p<0)
 {
  printf("Process creation failed\n");
 }
 else if (p==0)
 {
  printf("Child process created:%d\n",getpid());
 }
 else
 {
  printf("Parent process created:%d\n",getppid());
 }
 return(0);
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit procrefork.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc procrefork.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Parent process created:2898
Child process created:3123
```

EXP 5: FAMILIARIZATION OF wait() & sleep() SYSTEM CALLS

AIM: To familiarise the wait() and sleep() system calls.

PROGRAM

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
int main()
{
 int p=fork();
 if (p<0)
 {
  printf("Process creation failed\n");
 }
 else if (p==0)
 {
  sleep(5);
  printf("Child process is sleeping\n");
 }
 else
 {
  wait(NULL);
  printf("Parent process waiting\n");
 }
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit prowaitseek.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc  prowaitseek.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Child process is sleeping
Parent process waiting
```

EXP 6: FAMILIARIZATION OF execl() SYSTEM CALL

AIM: To familiarise with execl() system calls for
(i) listing the running processes
(ii) listing the files

PROGRAM 1

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
int main()
{
 execl("/bin/ps","ps",NULL);
 return(0);
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit proexec.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc  proexec.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
    PID TTY          TIME CMD
   2898 pts/2    00:00:00 bash
   3398 pts/2    00:00:00 ps
```

PROGRAM 2

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
int main()
{
 execlp("ls","ls","-l",NULL);
 return(0);
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit proexec.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit proexeclp.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc  proexeclp.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
total 1292
-rwxr-xr-x 1 cs20222066 CS2022   15968 Feb 24 13:23 a.out
-rwxr-xr-x 1 cs20222066 CS2022     313 Jan 27 13:59 arithmetic.sh
-rwxr-xr-x 1 cs20222066 CS2022     160 Feb 10 15:14 charoccur.sh
-rwxr-xr-x 1 cs20222066 CS2022     130 Feb 10 13:12 concatstr.sh
-rwxr-xr-x 1 cs20222066 CS2022     113 Jan 27 13:58 evenodd.sh
-rw-r--r-- 1 cs20222066 CS2022    3223 Jan 20 15:45 Exp_1C.txt
-rwxr-xr-x 1 cs20222066 CS2022     149 Jan 27 14:02 factorial.sh
-rwxr-xr-x 1 cs20222066 CS2022     165 Jan 27 14:14 fibonacci.sh
-rwxr-xr-x 1 cs20222066 CS2022     203 Jan 27 13:59 large3.sh
-rw-r--r-- 1 cs20222066 CS2022 1192719 Feb 10 15:31 OS_FAIR.odt
-rwxr-xr-x 1 cs20222066 CS2022     235 Jan 27 14:18 palindnum.sh
-rwxr-xr-x 1 cs20222066 CS2022     141 Feb 10 14:19 palindstr.sh
drwxr-xr-x 4 cs20222066 CS2022    4096 Jan 27 14:39 PHOTOS
-rwxr-xr-x 1 cs20222066 CS2022     261 Jan 27 14:07 prime.sh
-rw-r--r-- 1 cs20222066 CS2022     314 Feb 24 13:07 procrefork.c
-rw-r--r-- 1 cs20222066 CS2022     140 Feb 24 13:19 proexec.c
-rw-r--r-- 1 cs20222066 CS2022     141 Feb 24 13:23 proexeclp.c
-rw-r--r-- 1 cs20222066 CS2022     308 Feb 24 13:15 prowaitseek.c
-rwxr-xr-x 1 cs20222066 CS2022     121 Feb 10 14:22 remleadtrailsp.sh
-rwxr-xr-x 1 cs20222066 CS2022     182 Feb 10 14:09 replstr.sh
-rwxr-xr-x 1 cs20222066 CS2022      72 Feb 10 13:10 revstring.sh
-rwxr-xr-x 1 cs20222066 CS2022     274 Jan 27 14:24 sortarray.sh
-rwxr-xr-x 1 cs20222066 CS2022     171 Feb 10 15:15 strequal.sh
-rwxr-xr-x 1 cs20222066 CS2022      73 Feb 10 14:04 strupper.sh
-rwxr-xr-x 1 cs20222066 CS2022     213 Feb 10 14:25 substrpos.sh
-rwxr-xr-x 1 cs20222066 CS2022     169 Jan 27 14:10 sum.sh
-rwxr-xr-x 1 cs20222066 CS2022     138 Jan 27 13:54 swap.sh
-rwxr-xr-x 1 cs20222066 CS2022     161 Feb 10 15:15 wordoccurfile.sh
-rw-r--r-- 1 cs20222066 CS2022      97 Feb 10 15:15 wordoccurfile.txt
```

EXP 8: COPY A FILE TO A NEW FILE USING SYSTEM CALLS

AIM: To write a C program to copy files to a new file using system calls

PROGRAM

```
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
int main(int argc,char*argv[])
{
 char c;
 int fd1,fd2;
 ssize_t readit;
 if (argc>=4)
 {
  printf("Error");
 }
 else
 {
  fd1=open(argv[1],O_RDONLY);
  if (fd1==-1)
  {
   perror("Error!Cannot open fd1");
   return 0;
  }
  fd2=open(argv[2],O_WRONLY|O_CREAT,0644);
  if (fd2==-1)
  {
   perror("Error!Cannot open fd2");
   close(fd1);
   return 0;
  }
  while ((readit=read(fd1,&c,1))>0)
  {
   write(fd2,&c,1);
  }
  printf("Copied");
  close(fd1);
  close(fd2);
  return 0;
 }
}
```

OUTPUT

Open arithmetic.sh ~/Desktop/MEENAKSHI_B2_62 Save

```
echo "Enter 2 numbers(space-separated):"
read a b
zero=0
echo "Enter operator(1:+,2:-,3:*,4:/):"
read op
case $op in
        1)echo "Sum:$((a+b))";;
        2)echo "Difference:$((a-b))";;
        3)echo "Product:$((a*b))";;
        4)if [ $b -ne $zero ]; then
                echo "Quotient:$((a/b))"
          else
                echo "Division not possible!"
          fi
esac
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit filcopynew.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc filcopynew.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out arithmetic.sh arithmeticcopy.sh
Copiedcs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ cat arithmeticcopy.sh
echo "Enter 2 numbers(space-separated):"
read a b
zero=0
echo "Enter operator(1:+,2:-,3:*,4:/):"
read op
case $op in
        1)echo "Sum:$((a+b))";;
        2)echo "Difference:$((a-b))";;
        3)echo "Product:$((a*b))";;
        4)if [ $b -ne $zero ]; then
                echo "Quotient:$((a/b))"
          else
                echo "Division not possible!"
          fi
esac
```

EXP 9: COPY FILES IN REVERSE ORDER

AIM: To write a C program to copy files in a reverse order to a new file using system calls

PROGRAM

```
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc,char*argv[])
{
 char c;
 int fd1,fd2;
 int i=0;
 if (argc>=4)
 {
  printf("Error");
 }
 else
 {
  fd1=open(argv[1],O_RDONLY);
  if (fd1==-1)
  {
   perror("Error!Cannot open fd1");
   exit(EXIT_FAILURE);
  }
  fd2=open(argv[2],O_WRONLY|O_CREAT,0666);
  if (fd2==-1)
  {
   perror("Error!Cannot open fd2");
   close(fd1);
   exit(EXIT_FAILURE);
  }
  int start=lseek(fd1,0,SEEK_CUR);
  int end=lseek(fd1,0,SEEK_END);
  int restart=lseek(fd1,0-end,SEEK_CUR);
  char data[end];
  read(fd1,data,end);
  for (i=0;i<end;i++)
  {
   write(fd2,&data[end-(i+1)],1);
  }
  close(fd1);
  close(fd2);
  return 0;
 }
}
```

OUTPUT

arithmetic.sh
~/Desktop/MEENAKSHI_B2_62

```
echo "Enter 2 numbers(space-separated):"
read a b
zero=0
echo "Enter operator(1:+,2:-,3:*,4:/):"
read op
case $op in
        1)echo "Sum:$((a+b))";;
        2)echo "Difference:$((a-b))";;
        3)echo "Product:$((a*b))";;
        4)if [ $b -ne $zero ]; then
                echo "Quotient:$((a/b))"
          else
                echo "Division not possible!"
          fi
esac
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gedit filcopyrev.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc filcopyrev.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out arithmetic.sh arithrevcopy.sh
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ cat arithrevcopy.sh

case
if
"!elbissop ton noisiviD" ohce
esle
"))b/a(($:tneitouQ" ohce
neht ;] orez$ en- b$ [ fi)4
;;"))b*a(($:tcudorP" ohce)3
;;"))b-a(($:ecnereffiD" ohce)2
;;"))b+a(($:muS" ohce)1
 ni po$ esac
po daer
":)/:4,*:3,-:2,+:1(rotarepo retnE" ohce
0=orez
b a daer
```

EXP 10: FCFS SCHEDULING ALGORITHM

AIM: Implementing First Come First Serve(FCFS) process scheduling in C program and diplay details.

PROGRAM

```
#include <stdio.h>
typedef struct
{
 int pid,at,bt,ct,tat,wt;
} Process;

void sortByArrival(Process p[],int n)
{
 for (int i=0;i<n-1;i++)
  for (int j=0;j<n-i-1;j++)
   if (p[j].at > p[j+1].at)
   {
    Process temp=p[j];
    p[j]=p[j+1];
    p[j+1]=temp;
   }
}

void FCFS(Process p[],int n)
{
 sortByArrival(p,n);
 int time=0;
 printf("\nGantt Chart: ");
 for (int i=0; i<n; i++)
 {
  if (time <p[i].at) time=p[i].at;
  p[i].ct = time+p[i].bt;
  time=p[i].ct;
  printf("|P%d",p[i].pid);
 }
 printf("|\n");
 printf("\nPID\tAT\tBT\tCT\tTAT\tWT\n");
 float totalTAT=0, totalWT=0;
 for (int i=0; i<n;i++)
 {
  p[i].tat=p[i].ct-p[i].at;
  p[i].wt=p[i].tat-p[i].bt;
  totalTAT+=p[i].tat;
  totalWT+=p[i].wt;
  printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
 }
 printf("\nAverage TAT:%.2f", totalTAT/n);
 printf("\nAverage WT:%.2f\n",totalWT/n);
```

```
}

int main()
{
 int n;
 printf("Enter number of processes:");
 scanf("%d",&n);
 Process p[n];
 for (int i=0; i<n; i++)
 {
  printf("Enter AT, BT for P%d: ",i+1);
  scanf("%d%d", &p[i].at, &p[i].bt);
  p[i].pid=i+1;
 }
 FCFS(p,n);
 return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc fcfsalgo.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:3
Enter AT, BT for P1: 0 24
Enter AT, BT for P2: 0 3
Enter AT, BT for P3: 0 3

Gantt Chart: |P1|P2|P3|

PID     AT      BT      CT      TAT     WT
1       0       24      24      24      0
2       0       3       27      27      24
3       0       3       30      30      27

Average TAT:27.00
Average WT:17.00
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc fcfsalgo.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:4
Enter AT, BT for P1: 1 2
Enter AT, BT for P2: 2 1
Enter AT, BT for P3: 3 4
Enter AT, BT for P4: 4 5

Gantt Chart: |P1|P2|P3|P4|

PID     AT      BT      CT      TAT     WT
1       1       2       3       2       0
2       2       1       4       2       1
3       3       4       8       5       1
4       4       5       13      9       4

Average TAT:4.50
Average WT:1.50
```

EXP 11: SJF SCHEDULING ALGORITHM

AIM: Implementing Shortest Job First(SJF) process scheduling in C program and diplay details.

PROGRAM

```
#include <stdio.h>
#include <limits.h>
typedef struct
{
 int pid,at,bt,ct,tat,wt;
} Process;

void SJF(Process p[],int n)
{
 int time=0,completed=0, minIndex;
 printf("SJF");
 printf("\nGantt Chart: ");
 while (completed < n)
 {
  int minBT=INT_MAX;
  minIndex=-1;
  for (int i=0;i<n;i++)
  {
   if (p[i].at <= time && p[i].ct == 0 && p[i].bt < minBT)
   {
    minBT=p[i].bt;
    minIndex=i;
   }
  }
  if (minIndex==-1)
  {
   time++;
   continue;
  }
  time+=p[minIndex].bt;
  p[minIndex].ct=time;
  completed++;
  printf("|P%d", p[minIndex].pid);
 }
 printf("|\n");
 printf("\nPID\tAT\tBT\tCT\tTAT\tWT\n");
 float totalTAT=0, totalWT=0;
 for (int i=0; i<n;i++)
 {
  p[i].tat=p[i].ct-p[i].at;
  p[i].wt=p[i].tat-p[i].bt;
  totalTAT+=p[i].tat;
  totalWT+=p[i].wt;
```

```
  printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
 }
 printf("\nAverage TAT:%.2f", totalTAT/n);
 printf("\nAverage WT:%.2f\n",totalWT/n);
}

int main()
{
 int n;
 printf("Enter number of processes:");
 scanf("%d",&n);
 Process p[n];
 for (int i=0; i<n; i++)
 {
  printf("Enter AT, BT for P%d: ",i+1);
  scanf("%d%d", &p[i].at, &p[i].bt);
  p[i].pid=i+1;
 }
 SJF(p,n);
 return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc sjfalgo.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:3
Enter AT, BT for P1: 0 5
Enter AT, BT for P2: 2 9
Enter AT, BT for P3: 1 3

Gantt Chart: |P1|P3|P2|

PID     AT      BT      CT      TAT     WT
1       0       5       5       5       0
2       2       9       17      15      6
3       1       3       8       7       4

Average TAT:9.00
Average WT:3.33
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:5
Enter AT, BT for P1: 0 5
Enter AT, BT for P2: 2 9
Enter AT, BT for P3: 1 3
Enter AT, BT for P4: 3 5
Enter AT, BT for P5: 2 2

Gantt Chart: |P1|P5|P3|P4|P2|

PID     AT      BT      CT      TAT     WT
1       0       5       5       5       0
2       2       9       24      22      13
3       1       3       10      9       6
4       3       5       15      12      7
5       2       2       7       5       3

Average TAT:10.60
Average WT:5.80
```

EXP 12: SRTF SCHEDULING ALGORITHM

AIM: Implementing Shortest Remaining Time First(SRTF) process scheduling in C program and diplay details.

PROGRAM

```
#include <stdio.h>
#include <limits.h>
typedef struct
{
 int pid,at,bt,ct,tat,wt,remaining_bt;
} Process;

void SRTF(Process p[],int n)
{
 int time=0,completed=0,minIndex,minBT;
 for (int i=0;i<n;i++)
  p[i].remaining_bt=p[i].bt;
 printf("\nGantt Chart: ");
 while (completed <n)
 {
  minBT=INT_MAX;
  minIndex=-1;
  for (int i=0;i<n;i++)
  {
   if (p[i].at <= time && p[i].remaining_bt>0 && p[i].remaining_bt < minBT)
   {
    minBT=p[i].remaining_bt;
    minIndex=i;
   }
  }
  if (minIndex==-1)
  {
   time++;
   continue;
  }
  printf("|P%d", p[minIndex].pid);
  p[minIndex].remaining_bt--;
  time++;
  if (p[minIndex].remaining_bt==0)
  {
   p[minIndex].ct=time;
   completed++;
  }
 }
 printf("|\n");
 printf("\nPID\tAT\tBT\tCT\tTAT\tWT\n");
```

```
 float totalTAT=0, totalWT=0;
 for (int i=0; i<n;i++)
 {
  p[i].tat=p[i].ct-p[i].at;
  p[i].wt=p[i].tat-p[i].bt;
  totalTAT+=p[i].tat;
  totalWT+=p[i].wt;
  printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
 }
 printf("\nAverage TAT:%.2f", totalTAT/n);
 printf("\nAverageWT:%.2f\n",totalWT/n);
}

int main()
{
 int n;
 printf("Enter number of processes:");
 scanf("%d",&n);
 Process p[n];
 for (int i=0; i<n; i++)
 {
  printf("Enter AT, BT for P%d: ",i+1);
  scanf("%d%d", &p[i].at, &p[i].bt);
  p[i].pid=i+1;
 }
 SRTF(p,n);
 return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc srtfalgo.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:4
Enter AT, BT for P1: 0 7
Enter AT, BT for P2: 2 4
Enter AT, BT for P3: 4 1
Enter AT, BT for P4: 5 4

Gantt Chart: |P1|P1|P2|P2|P3|P2|P2|P4|P4|P4|P4|P1|P1|P1|P1|P1|

PID      AT       BT       CT       TAT      WT
1        0        7        16       16       9
2        2        4        7        5        1
3        4        1        5        1        0
4        5        4        11       6        2

Average TAT:7.00
AverageWT:3.00
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes:4
Enter AT, BT for P1: 1 1
Enter AT, BT for P2: 4 3
Enter AT, BT for P3: 3 5
Enter AT, BT for P4: 2 3

Gantt Chart: |P1|P4|P4|P4|P2|P2|P2|P3|P3|P3|P3|P3|

PID      AT       BT       CT       TAT      WT
1        1        1        2        1        0
2        4        3        8        4        1
3        3        5        13       10       5
4        2        3        5        3        0

Average TAT:4.50
AverageWT:1.50
```

EXP 13: RR SCHEDULING ALGORITHM

AIM: Implementing Round Robin(RR) process scheduling in C program and diplay details.

PROGRAM

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

void queueUpdation(int queue[], int timer, int arrival[], int n, int maxProccessIndex) {
    int zeroIndex;
    for (int i = 0; i < n; i++) {
        if (queue[i] == 0) {
            zeroIndex = i;
            break;
        }
    }
    queue[zeroIndex] = maxProccessIndex + 1;
}

void queueMaintainence(int queue[], int n) {
    for (int i = 0; (i < n - 1) && (queue[i + 1] != 0); i++) {
        int temp = queue[i];
        queue[i] = queue[i + 1];
        queue[i + 1] = temp;
    }
}

void checkNewArrival(int timer, int arrival[], int n, int *maxProccessIndex, int queue[]) {
    if (timer <= arrival[n - 1]) {
        bool newArrival = false;
        for (int j = (*maxProccessIndex + 1); j < n; j++) {
            if (arrival[j] <= timer) {
                if (*maxProccessIndex < j) {
                    *maxProccessIndex = j;
                    newArrival = true;
                }
            }
        }
        if (newArrival)
            queueUpdation(queue, timer, arrival, n, *maxProccessIndex);
    }
}

void printGanttChart(int process[], int startTime[], int endTime[], int count) {
    printf("\nGantt Chart:\n");

    for (int i = 0; i < count; i++) {
        if (startTime[i] != endTime[i]) {
```

```
            printf("| P%d ", process[i]);
        }
    }
    printf("|");

    printf("\n%d", startTime[0]);
    for (int i = 0; i < count; i++) {
        if (startTime[i] != endTime[i]) {
            printf("\t%d", endTime[i]);
        }
    }
    printf("\n");
}

int main() {
    int n, tq, timer = 0, maxProccessIndex = 0;
    float avgWait = 0, avgTT = 0;

    printf("\nEnter the time quanta: ");
    scanf("%d", &tq);
    printf("\nEnter the number of processes: ");
    scanf("%d", &n);

    int arrival[n], burst[n], wait[n], turn[n], queue[n], temp_burst[n];
    bool complete[n];

    printf("\nEnter the arrival time of the processes: ");
    for (int i = 0; i < n; i++)
        scanf("%d", &arrival[i]);

    printf("\nEnter the burst time of the processes: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &burst[i]);
        temp_burst[i] = burst[i];
    }

    for (int i = 0; i < n; i++) {
        complete[i] = false;
        queue[i] = 0;
    }

    int process[1000], startTime[1000], endTime[1000], count = 0;

    while (timer < arrival[0])
        timer++;
    queue[0] = 1;

    while (1) {
        bool flag = true;
        for (int i = 0; i < n; i++) {
            if (temp_burst[i] != 0) {
                flag = false;
```

```
            break;
        }
    }
    if (flag)
        break;

    for (int i = 0; (i < n) && (queue[i] != 0); i++) {
        int ctr = 0;
        int start = timer;
        while ((ctr < tq) && (temp_burst[queue[0] - 1] > 0)) {
            temp_burst[queue[0] - 1] -= 1;
            timer += 1;
            ctr++;

            checkNewArrival(timer, arrival, n, &maxProccessIndex, queue);
        }

        process[count] = queue[0];
        startTime[count] = start;
        endTime[count] = timer;
        count++;

        if ((temp_burst[queue[0] - 1] == 0) && (complete[queue[0] - 1] == false)) {
            turn[queue[0] - 1] = timer;
            complete[queue[0] - 1] = true;
        }

        bool idle = true;
        if (queue[n - 1] == 0) {
            for (int i = 0; i < n && queue[i] != 0; i++) {
                if (complete[queue[i] - 1] == false) {
                    idle = false;
                }
            }
        } else
            idle = false;

        if (idle) {
            timer++;
            checkNewArrival(timer, arrival, n, &maxProccessIndex, queue);
        }

        queueMaintainence(queue, n);
    }
}

for (int i = 0; i < n; i++) {
    turn[i] = turn[i] - arrival[i];
    wait[i] = turn[i] - burst[i];
}

printf("\nProgram No.\tArrival Time\tBurst Time\tWait Time\tTurnAround Time\n");
```

```
    for (int i = 0; i < n; i++) {
        printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, arrival[i], burst[i], wait[i], turn[i]);
    }

    for (int i = 0; i < n; i++) {
        avgWait += wait[i];
        avgTT += turn[i];
    }

    printf("\nAverage wait time: %.2f\nAverage Turn Around Time: %.2f\n", (avgWait / n), (avgTT /
n));

    printGanttChart(process, startTime, endTime, count);

    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc rralgo.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out

Enter the time quanta: 2

Enter the number of processes: 4

Enter the arrival time of the processes: 0 1 3 5

Enter the burst time of the processes: 3 3 2 4

Program No.     Arrival Time    Burst Time      Wait Time       TurnAround Time
1               0               3               2               5
2               1               3               4               7
3               3               2               2               4
4               5               4               3               7

Average wait time: 2.75
Average Turn Around Time: 5.75

Gantt Chart:
| P1 | P2 | P1 | P3 | P2 | P4 | P4 |
0       2       4       5       7       8       10      12
```

EXP 14: PRIORITY SCHEDULING ALGORITHM

AIM: Implementing Priority Scheduling(Both Preemptive & Non-Preemptive) process scheduling in C program and diplay details.

PROGRAM

PREEMPTIVE:

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

struct process
{
   int pid, arrtime, burtime, remtime, comtime, waitime, turtime, priority;
};

void sortp(struct process p[], int n)
{
   for (int i = 0; i < n - 1; i++)
   {
      for (int j = 0; j < n - i - 1; j++)
      {
         if (p[j].arrtime > p[j + 1].arrtime)
         {
            struct process temp = p[j];
            p[j] = p[j + 1];
            p[j + 1] = temp;
         }
      }
   }
}

void calc(struct process p[], int n, bool smallerIsHigher)
{
   int curtime = 0, complete = 0;
   int lastp = -1;
   printf("\nGantt Chart:\n|");

   int timestamps[1000];
   int tsIndex = 0;

   while (complete < n)
   {
      int min = -1;
      int hpriority = smallerIsHigher ? INT_MAX : INT_MIN;

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

```
        {
            if (p[i].remtime > 0 && p[i].arrtime <= curtime)
            {
                if ((smallerIsHigher && p[i].priority < hpriority) || (!smallerIsHigher && p[i].priority >
hpriority))
                {
                    min = i;
                    hpriority = p[i].priority;
                }
            }
        }

        if (min == -1)
        {
            curtime++;
            continue;
        }

        if (p[min].pid != lastp)
        {
            printf(" P%d |", p[min].pid);
            timestamps[tsIndex++] = curtime;
            lastp = p[min].pid;
        }

        p[min].remtime--;
        curtime++;

        if (p[min].remtime == 0)
        {
            p[min].comtime = curtime;
            p[min].turtime = p[min].comtime - p[min].arrtime;
            p[min].waitime = p[min].turtime - p[min].burtime;
            complete++;
        }
    }
    timestamps[tsIndex++] = curtime;
    printf("\n");

    printf(" ");
    for (int i = 0; i < tsIndex; i++)
    {
        printf("%d ", timestamps[i]);
        if (i < tsIndex - 1)
            printf("   ");
    }
    printf("\n");
}

void disp(struct process p[], int n)
{
    float waitavg = 0, turnavg = 0;
```

```
    printf("\nPID\tArrival\tBurst\tPriority\tCompletion\tTurnaround\tWaiting\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d\t%d\t%d\t%d\t\t%d\t\t%d\t\t%d\n",
                p[i].pid, p[i].arrtime, p[i].burtime, p[i].priority, p[i].comtime, p[i].turtime, p[i].waitime);
        waitavg += p[i].waitime;
        turnavg += p[i].turtime;
    }
    printf("\nAverage Turnaround Time: %.2f\n", turnavg / n);
    printf("Average  Waiting  Time : %.2f\n", waitavg / n);
}

int main()
{
    int n, choice;
    printf("PREEMPTIVE PRIORITY SCHEDULING\n");
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    struct process p[n];

    for (int i = 0; i < n; i++)
    {
        printf("Arrival time for Process %d: ", i + 1);
        scanf("%d", &p[i].arrtime);
        printf("Burst  time for  Process %d: ", i + 1);
        scanf("%d", &p[i].burtime);
        printf("Priority for Process %d: ", i + 1);
        scanf("%d", &p[i].priority);

        p[i].pid = i + 1;
        p[i].remtime = p[i].burtime;
    }

    printf("\nChoose Priority Type:\n1. Smaller number = Higher Priority\n2. Larger number =
Higher Priority\nEnter choice: ");
    scanf("%d", &choice);

    bool smallerIsHigher = (choice == 1);
    sortp(p, n);
    calc(p, n, smallerIsHigher);
    disp(p, n);

    return 0;
}
```

NON-PREEMPTIVE:

```
#include <stdio.h>


struct Process {
```

```
    int id;
    int arrivalTime;
    int burstTime;
    int priority;
    int completionTime;
    int turnaroundTime;
    int waitingTime;
};

void calculateTimes(struct Process p[], int n, int isHigherPriorityBetter) {
    int currentTime = 0, completed = 0;
    int isCompleted[n];

    for (int i = 0; i < n; i++) {
        isCompleted[i] = 0;
    }

    printf("\nGantt Chart: \n");

    while (completed != n) {
        int bestPriority = isHigherPriorityBetter ? -1 : 1e9, index = -1;

        for (int i = 0; i < n; i++) {
            if (p[i].arrivalTime <= currentTime && !isCompleted[i]) {
                if ((isHigherPriorityBetter && p[i].priority > bestPriority) ||
                    (!isHigherPriorityBetter && p[i].priority < bestPriority)) {
                    bestPriority = p[i].priority;
                    index = i;
                }
            }
        }

        if (index == -1) {
            currentTime++;
        } else {
            printf("| P%d (%d) ", p[index].id, currentTime);
            currentTime += p[index].burstTime;
            p[index].completionTime = currentTime;
            p[index].turnaroundTime = p[index].completionTime - p[index].arrivalTime;
            p[index].waitingTime = p[index].turnaroundTime - p[index].burstTime;
            isCompleted[index] = 1;
            completed++;
        }
    }
    printf("| (%d)\n", currentTime);
}

void displayResults(struct Process p[], int n) {
    float totalTurnaroundTime = 0, totalWaitingTime = 0;

    printf("\nProcess\tArrival Time\tBurst Time\tPriority\tCompletion Time\tTurnaround Time\t
Waiting Time\n");
```

```
    for (int 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\t\t\n", p[i].id, p[i].arrivalTime,
p[i].burstTime, p[i].priority, p[i].completionTime, p[i].turnaroundTime, p[i].waitingTime);
        totalTurnaroundTime += p[i].turnaroundTime;
        totalWaitingTime += p[i].waitingTime;
    }

    printf("\nAverage Turnaround Time: %.2f", totalTurnaroundTime / n);
    printf("\nAverage Waiting Time: %.2f\n", totalWaitingTime / n);
}

int main() {
    int n, isHigherPriorityBetter;
    printf("Enter number of processes: ");
    scanf("%d", &n);

    printf("Enter 1 if larger numbers have higher priority, or 0 if smaller numbers have higher
priority: ");
    scanf("%d", &isHigherPriorityBetter);

    struct Process p[n];

    for (int i = 0; i < n; i++) {
        p[i].id = i + 1;
        printf("Enter Arrival Time, Burst Time, and Priority for Process %d: ", i + 1);
        scanf("%d %d %d", &p[i].arrivalTime, &p[i].burstTime, &p[i].priority);
    }

    calculateTimes(p, n, isHigherPriorityBetter);
    displayResults(p, n);

    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc priornonpre.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter number of processes: 4
Enter 1 if larger numbers have higher priority, or 0 if smaller numbers have higher priority: 0
Enter Arrival Time, Burst Time, and Priority for Process 1: 0 4 3
Enter Arrival Time, Burst Time, and Priority for Process 2: 1 2 2
Enter Arrival Time, Burst Time, and Priority for Process 3: 2 3 4
Enter Arrival Time, Burst Time, and Priority for Process 4: 4 2 1

Gantt Chart:
| P1 (0) | P4 (4) | P2 (6) | P3 (8) | (11)

Process Arrival Time    Burst Time      Priority        Completion Time Turnaround Time  Waiting Time
P1              0               4               3               4               4               0
P2              1               2               2               8               7               5
P3              2               3               4               11              9               6
P4              4               2               1               6               2               0

Average Turnaround Time: 5.50
Average Waiting Time: 2.75
```

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc priorpre.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
PREEMPTIVE PRIORITY SCHEDULING
Enter the number of processes: 4
Arrival time for Process 1: 0
Burst  time for  Process 1: 4
Priority for Process 1: 3
Arrival time for Process 2: 1
Burst  time for  Process 2: 2
Priority for Process 2: 2
Arrival time for Process 3: 2
Burst  time for  Process 3: 3
Priority for Process 3: 4
Arrival time for Process 4: 4
Burst  time for  Process 4: 2
Priority for Process 4: 1

Choose Priority Type:
1. Smaller number = Higher Priority
2. Larger number = Higher Priority
Enter choice: 1

Gantt Chart:
| P1 | P2 | P1 | P4 | P1 | P3 |
 0    1    3    4    6    8    11

PID     Arrival Burst   Priority        Completion      Turnaround      Waiting
1       0       4       3               8               8               4
2       1       2       2               3               2               0
3       2       3       4               11              9               6
4       4       2       1               6               2               0

Average Turnaround Time: 5.25
Average  Waiting  Time : 2.50
```

EXP 15: FIFO PAGE REPLACEMENT ALGORITHM

AIM: Implementing First In First Out(FIFO) page replacement algorithm in C program and display details.

PROGRAM

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

int *queue, front = -1, rear = -1, SIZE, pageFaults = 0, totalInputs = 0;

int isEmpty() {
    return front == -1;
}

int isFull() {
    return (rear + 1) % SIZE == front;
}

int exists(int value) {
    if (isEmpty()) return 0;
    for (int i = front; ; i = (i + 1) % SIZE) {
        if (queue[i] == value) {
            return 1;
        }
        if (i == rear) break;
    }
    return 0;
}

void enqueue(int value) {
    totalInputs++;
    if (exists(value)) {
        return;
    }
    pageFaults++;
    if (isFull()) {
        front = (front + 1) % SIZE;
    }
    if (isEmpty()) {
        front = rear = 0;
    } else {
        rear = (rear + 1) % SIZE;
    }
    queue[rear] = value;
```

```
}


void display() {
    if (isEmpty()) {
        printf("Queue is empty!\n");
        return;
    }
    printf("Final Queue: ");
    for (int i = front; ; i = (i + 1) % SIZE) {
        printf("%d ", queue[i]);
        if (i == rear) break;
    }
    printf("\n");
}

int main() {
    printf("Enter the frame size of the queue: ");
    scanf("%d", &SIZE);
    queue = (int *)malloc(SIZE * sizeof(int));
    if (queue == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter page reference values separated by spaces (press ENTER to finish): ");
    while (getchar() != '\n');

    char input[1024];
    fgets(input, sizeof(input), stdin);

    char *token = strtok(input, " ");
    while (token != NULL) {
        int value = atoi(token);
        enqueue(value);
        token = strtok(NULL, " ");
    }

    display();
    printf("Total Page Faults: %d\n", pageFaults);
    if (totalInputs > 0) {
        float hitRatio = (float)(totalInputs - pageFaults) / totalInputs;
        float missRatio = (float)pageFaults / totalInputs;
        printf("Hit Ratio: %.2f\n", hitRatio);
        printf("Miss Ratio: %.2f\n", missRatio);
    }

    free(queue);
    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc fifopage.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter the frame size of the queue: 4
Enter page reference values separated by spaces (press ENTER to finish): 0 1 2 3 0 2 4 0 4 2 0 5 4 0 1
Final Queue: 4 0 5 1
Total Page Faults: 8
Hit Ratio: 0.47
Miss Ratio: 0.53
```

EXP 16: OPR PAGE REPLACEMENT ALGORITHM

AIM: Implementing Optical Page Replacement(OPR) in C program and display details.

PROGRAM

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

int *pageReferences, *pageFrames, *usage;
int NUM_FRAMES, NUM_PAGES, pageFaults = 0, pageHits = 0, totalInputs = 0;


int findOptimalPage(int index) {
    int farthestIndex = -1, farthest = -1;
    for (int i = 0; i < NUM_FRAMES; i++) {
        int j;
        for (j = index; j < NUM_PAGES; j++) {
            if (pageFrames[i] == pageReferences[j]) {
                if (j > farthest) {
                    farthest = j;
                    farthestIndex = i;
                }
                break;
            }
        }
        if (j == NUM_PAGES) {
            return i;
        }
    }
    return (farthestIndex == -1) ? 0 : farthestIndex;
}


int exists(int value) {
    for (int i = 0; i < NUM_FRAMES; i++) {
        if (pageFrames[i] == value) {
            usage[i] = totalInputs;
            return i;
        }
    }
    return -1;
}

void processPages() {
    for (int i = 0; i < NUM_PAGES; i++) {
        totalInputs++;
        int page = pageReferences[i];
```

```
        int index = exists(page);

        if (index != -1) {
            pageHits++;
            continue;
        }

        pageFaults++;
        int replaceIndex = -1;
        for (int j = 0; j < NUM_FRAMES; j++) {
            if (pageFrames[j] == -1) {
                replaceIndex = j;
                break;
            }
        }
        if (replaceIndex == -1) {
            replaceIndex = findOptimalPage(i + 1);
        }
        pageFrames[replaceIndex] = page;
        usage[replaceIndex] = totalInputs;
    }
}


void displayResults() {
    printf("Final Frames: ");
    for (int i = 0; i < NUM_FRAMES; i++) {
        printf("%d ", pageFrames[i]);
    }
    printf("\nTotal Page Faults: %d\n", pageFaults);
    float hitRatio = (float)pageHits / totalInputs;
    float missRatio = (float)pageFaults / totalInputs;
    printf("Hit Ratio: %.2f\nMiss Ratio: %.2f\n", hitRatio, missRatio);
}

int main() {
    printf("Enter the frame size: ");
    scanf("%d", &NUM_FRAMES);
    printf("Enter the number of pages: ");
    scanf("%d", &NUM_PAGES);

    pageReferences = malloc(NUM_PAGES * sizeof(int));
    pageFrames = malloc(NUM_FRAMES * sizeof(int));
    usage = malloc(NUM_FRAMES * sizeof(int));
    if (!pageReferences || !pageFrames || !usage) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    memset(pageFrames, -1, NUM_FRAMES * sizeof(int));

    printf("Enter the page reference string (space-separated): ");
```

```
    for (int i = 0; i < NUM_PAGES; i++) {
        scanf("%d", &pageReferences[i]);
    }

    processPages();
    displayResults();

    free(pageReferences);
    free(pageFrames);
    free(usage);
    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc oprpage.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter the frame size: 4
Enter the number of pages: 15
Enter the page reference string (space-separated): 0 1 2 3 0 2 4 0 4 2 0 5 4 0 1
Final Frames: 0 1 5 4
Total Page Faults: 6
Hit Ratio: 0.60
Miss Ratio: 0.40
```

EXP 17: LRU PAGE REPLACEMENT ALGORITHM

AIM: Implementing Least Recently Used(LRU) page replacement in C program and display details.

PROGRAM

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

int *queue, *usage, SIZE, pageFaults = 0, totalInputs = 0, count = 0;


int isFull() {
   return count == SIZE;
}


int isEmpty() {
   return count == 0;
}


int findLRU() {
   int lruIndex = 0, minUsage = usage[0];
   for (int i = 1; i < count; i++) {
      if (usage[i] < minUsage) {
         minUsage = usage[i];
         lruIndex = i;
      }
   }
   return lruIndex;
}


int exists(int value) {
   for (int i = 0; i < count; i++) {
      if (queue[i] == value) {
         usage[i] = totalInputs;
         return i;
      }
   }
   return -1;
}


void enqueue(int value) {
   totalInputs++;
```

```
    int index = exists(value);
    if (index != -1) {
        return;
    }
    pageFaults++;

    if (isFull()) {
        int lruIndex = findLRU();
        queue[lruIndex] = value;
        usage[lruIndex] = totalInputs;
        return;
    }

    queue[count] = value;
    usage[count] = totalInputs;
    count++;
}

void display() {
    if (isEmpty()) {
        printf("Queue is empty!\n");
        return;
    }
    printf("Final Queue: ");
    for (int i = 0; i < count; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}

int main() {
    printf("Enter the frame size of the queue: ");
    scanf("%d", &SIZE);
    queue = (int *)malloc(SIZE * sizeof(int));
    usage = (int *)malloc(SIZE * sizeof(int));
    if (queue == NULL || usage == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter page reference values separated by spaces (press ENTER to finish): ");
    while (getchar() != '\n');

    char input[1024];
    fgets(input, sizeof(input), stdin);

    char *token = strtok(input, " ");
    while (token != NULL) {
        int value = atoi(token);
        enqueue(value);
        token = strtok(NULL, " ");
    }
```

```
    display();
    printf("Total Page Faults: %d\n", pageFaults);
    if (totalInputs > 0) {
        float hitRatio = (float)(totalInputs - pageFaults) / totalInputs;
        float missRatio = (float)pageFaults / totalInputs;
        printf("Hit Ratio: %.2f\n", hitRatio);
        printf("Miss Ratio: %.2f\n", missRatio);
    }

    free(queue);
    free(usage);
    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc lrupage.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter the frame size of the queue: 4
Enter page reference values separated by spaces (press ENTER to finish): 0 1 2 3 0 2 4 0 4 2 0 5 4 0 1
Final Queue: 0 4 1 5
Total Page Faults: 7
Hit Ratio: 0.53
Miss Ratio: 0.47
```

EXP 18: MRU PAGE REPLACEMENT ALGORITHM

AIM: Implementing Most Recently Used(MRU) page replacement in C program and display details.

PROGRAM

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

int *queue, *timeStamp, front = -1, rear = -1, SIZE, pageFaults = 0, totalInputs = 0, currentTime =
0;

int isEmpty() {
   return front == -1;
}

int exists(int value) {
   if (isEmpty()) return -1;
   for (int i = front; ; i = (i + 1) % SIZE) {
      if (queue[i] == value) {
         return i;
      }
      if (i == rear) break;
   }
   return -1;
}

int findMRU() {
   int maxTime = -1, index = -1;
   for (int i = front; ; i = (i + 1) % SIZE) {
      if (timeStamp[i] > maxTime) {
         maxTime = timeStamp[i];
         index = i;
      }
      if (i == rear) break;
   }
   return index;
}

void enqueue(int value) {
   totalInputs++;
   int index = exists(value);
```

```
    if (index != -1) {
        timeStamp[index] = currentTime++;
        return;
    }

    pageFaults++;
    if (isEmpty()) {
        front = rear = 0;
    } else if ((rear + 1) % SIZE == front) {
        int mruIndex = findMRU();
        queue[mruIndex] = value;
        timeStamp[mruIndex] = currentTime++;
        return;
    } else {
        rear = (rear + 1) % SIZE;
    }
    queue[rear] = value;
    timeStamp[rear] = currentTime++;
}


void display() {
    if (isEmpty()) {
        printf("Queue is empty!\n");
        return;
    }
    printf("Final Queue: ");
    for (int i = front; ; i = (i + 1) % SIZE) {
        printf("%d ", queue[i]);
        if (i == rear) break;
    }
    printf("\n");
}

int main() {
    printf("Enter the frame size of the queue: ");
    scanf("%d", &SIZE);
    queue = (int *)malloc(SIZE * sizeof(int));
    timeStamp = (int *)malloc(SIZE * sizeof(int));

    if (queue == NULL || timeStamp == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }

    printf("Enter page reference values separated by spaces (press ENTER to finish): ");
    while (getchar() != '\n'); // Clear input buffer

    char input[1024];
    fgets(input, sizeof(input), stdin);

    char *token = strtok(input, " ");
```

```
    while (token != NULL) {
        int value = atoi(token);
        enqueue(value);
        token = strtok(NULL, " ");
    }

    display();
    printf("Total Page Faults: %d\n", pageFaults);
    if (totalInputs > 0) {
        float hitRatio = (float)(totalInputs - pageFaults) / totalInputs;
        float missRatio = (float)pageFaults / totalInputs;
        printf("Hit Ratio: %.2f\n", hitRatio);
        printf("Miss Ratio: %.2f\n", missRatio);
    }

    free(queue);
    free(timeStamp);
    return 0;
}
```

OUTPUT

```
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ gcc mrupage.c
cs20222066@NOSLAB108:~/Desktop/MEENAKSHI_B2_62$ ./a.out
Enter the frame size of the queue: 4
Enter page reference values separated by spaces (press ENTER to finish): 0 1 2 3 0 2 4 0 4 2 0 5 4 0 1
Final Queue: 0 1 2 3
Total Page Faults: 9
Hit Ratio: 0.40
Miss Ratio: 0.60
```