1. Write a Shell program to check the given number and its reverse are same.

echo "Enter a number:"

read num

rev=$(echo "$num" | rev)

if [ "$num" -eq "$rev" ]; then

echo "The number and its reverse are the same."

else

echo "The number and its reverse are not the same."

fi

2. Implementation of Non-Preemptive and Preemptive CPU Scheduling Algorithms.

1. Write a Shell Program to find a factorial of a number.

echo "Enter a number:"

read num

factorial=1

for ((i = 2; i <= num; i++)); do

factorial=$((factorial \* i))

done

echo "Factorial of $num is $factorial."

2. Implementation of Banker’s Algorithm for Deadlock Avoidance.

1. Write a Shell Program for finding sum of Odd and Even numbers up to ‘N’.

echo "Enter the value of N:"

read N

sum\_even=0

sum\_odd=0

for ((i = 1; i <= N; i++)); do

if ((i % 2 == 0)); then

sum\_even=$((sum\_even + i))

else

sum\_odd=$((sum\_odd + i))

fi

done

echo "Sum of Even numbers: $sum\_even"

echo "Sum of Odd numbers: $sum\_odd"

2. Implementation of Memory Allocation and Management Techniques.

1. Write a Shell Program to have to print half pyramid using for loop.

echo "Enter the number of rows:"

read rows

for ((i = 1; i <= rows; i++)); do

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

echo -n "\* "

done

echo

done

2. Implementation of Dining Philosopher’s Problem to demonstrate Process Synchronization.

1.Write a Shell program to display student grades

echo "Enter the marks of the student:"

read marks

if [ "$marks" -ge 90 ]; then

echo "Grade: A"

elif [ "$marks" -ge 80 ]; then

echo "Grade: B"

elif [ "$marks" -ge 70 ]; then

echo "Grade: C"

elif [ "$marks" -ge 60 ]; then

echo "Grade: D"

else

echo "Grade: F"

fi

2.Implementation of Page Replacement Techniques.

1.Write a Shell Program to perform Arithmetic Operation using Case statement.

echo "Enter two numbers:"

read a b

echo "Enter an operation (+, -, \*, /):"

read op

case "$op" in

"+") echo "Result: $((a + b))" ;;

"-") echo "Result: $((a - b))" ;;

"\*") echo "Result: $((a \* b))" ;;

"/") echo "Result: $((a / b))" ;;

\*) echo "Invalid operation" ;;

esac

2.Implementation of Memory Allocation and Management Techniques.

1. Write a Shell Program for comparison of strings.

echo "Enter the first string:"

read str1

echo "Enter the second string:"

read str2

if [ "$str1" = "$str2" ]; then

echo "Strings are equal."

else

echo "Strings are not equal."

Fi

1.Write a Shell program to find the smallest number from a set of numbers.

echo "Enter numbers separated by spaces:"

read -a numbers

smallest=${numbers[0]}

for num in "${numbers[@]}"; do

if [ "$num" -lt "$smallest" ]; then

smallest=$num

fi

done

echo "The smallest number is $smallest."

2. Implementation of Memory Allocation and Management Techniques.

1. Write a Shell program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5.

sum=0

for ((i = 50; i <= 100; i++)); do

if ((i % 3 == 0 && i % 5 != 0)); then

sum=$((sum + i))

fi

done

echo "The sum is $sum."

2.Implementation of Banker’s Algorithm for Deadlock Avoidance.