1) Write a Shell program to check the given number is even or odd.

program:

```
echo -n "Enter a number : "
read n
if [ `expr $n % 2` -eq 0 ]
then
   echo "$n is even"
else
   echo "$n is odd"
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./1.sh
Enter a number : 4
4 is even
```

2) Write a Shell program to check a leap year.

program:

```
echo -n "Enter year : "
read n
if [ `expr $n % 4` -eq 0 ] && [ `expr $n % 100` -ne 0 ] || [ `expr $n % 400` -eq 0 ];
then
   echo "$n is leap year"
else
   echo "$n is not a leap year"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./2.sh
Enter year : 2024
2024 is leap year
```

3) Write a Shell program to find the area and circumference of a circle.

program:

```
echo "Enter the radius of the circle:"
read radius
area=$(echo "3.1415 * ($radius ^ 2)" | bc)
circumference=$(echo "2 * 3.1415 * $radius" | bc)
echo "The area of the circle is: $area"
echo "The circumference of the circle is: $circumference"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./3.sh
Enter the radius of the circle:
3
The area of the circle is: 28.2735
The circumference of the circle is: 18.8490
```

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

program:

```
echo "Enter a number:"
read number
reverse=$(echo $number | rev)
if [ $number -eq $reverse ]; then
   echo "The number and its reverse are the same."
else
   echo "The number and its reverse are not the same."
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./4.sh
Enter a number:
1221
The number and its reverse are the same.
```

5) Write a Shell program to check the given string is palindrome or not.

program:

```
echo "Enter a string:"
read string
reverse=$(echo $string | rev)
if [ "$string" == "$reverse" ]; then
   echo "The string is a palindrome."
else
   echo "The string is not a palindrome."
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./5.sh
Enter a string:
hellolleh
The string is a palindrome.
```

6) Write a Shell program to find the sum of odd and even numbers from a set of numbers.

program:

```
echo "Enter numbers separated by space:"
read -a numbers
sum_even=0
sum_odd=0
for num in "${numbers[@]}"; do
    if [ $((num % 2)) -eq 0 ]; then
        sum_even=$((sum_even + num))
    else
        sum_odd=$((sum_odd + num))
    fi
done
echo "Sum of even numbers: $sum_even"
echo "Sum of odd numbers: $sum_odd"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./6.sh
Enter numbers separated by space:
10 11 12 13 14 15 16
Sum of even numbers: 52
Sum of odd numbers: 39
```

7) Write a Shell program to find the roots of a quadratic equation.

program:

```
echo "Enter the coefficients (a, b, c) of the quadratic equation (ax^2 + bx + c):"
read a b c
discriminant=$((b * b - 4 * a * c))
if [ $discriminant -gt 0 ]; then
   root1=$(echo "(-$b + sqrt($discriminant)) / (2 * $a)" | bc -l)
   root2=$(echo "(-$b - sqrt($discriminant)) / (2 * $a)" | bc -l)
   echo "The roots are real and different."
   echo "Root 1 = $root1, Root 2 = $root2"
elif [ $discriminant -eq 0 ]; then
   root=$(echo "-$b / (2 * $a)" | bc -l)
   echo "The roots are real and equal."
   echo "Root 1 = Root 2 = $root"
else
   real_part=$(echo "-$b / (2 * $a)" | bc -l)
   imaginary_part=$(echo "sqrt($((-1 * discriminant))) / (2 * $a)" | bc -l)
   echo "The roots are complex and different."
   echo "Root 1 = $real_part + $imaginary_part i"
   echo "Root 2 = $real_part - $imaginary_part i"
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./7.sh
Enter the coefficients (a, b, c) of the quadratic equation (ax^2 + bx + c):
1 10 2
The roots are real and different.
Root 1 = -.20416847668728045840, Root 2 = -9.79583152331271954159
```

8) Write a Shell program to check the given integer is Armstrong number or not.

program:

```
echo "Enter a number:"
read number
length=${#number}
sum=0
for ((i=0; i<$length; i++)); do
   digit=${number:i:1}
   sum=$((sum + digit ** length))
done
if [ $sum -eq $number ]; then
   echo "$number is an Armstrong number."
else
   echo "$number is not an Armstrong number."
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./8.sh
Enter a number:
153
153 is an Armstrong number.
```

9) Write a Shell program to check the given integer is prime or not.

program:

```
echo "Enter a number:"
read number
is_prime=true
if [ $number -lt 2 ]; then
   is_prime=false
fi
for ((i=2; i<=number/2; i++)); do
   if [ $((number % i)) -eq 0 ]; then
      is_prime=false
      break
   fi
done
if $is_prime; then
   echo "$number is a prime number."
else
   echo "$number is not a prime number."
fi
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./9.sh
Enter a number:
23
23 is a prime number.
```

10) Write a Shell program to generate prime numbers between 1 and 50.

program:

```
echo "Prime numbers between 1 and 50 are:"
for ((i=2; i<=50; i++)); do
    is_prime=true
    for ((j=2; j<=i/2; j++)); do
        if [ $((i % j)) -eq 0 ]; then
            is_prime=false
            break
        fi
    done
    if $is_prime; then
        echo $i
    fi
done
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./10.sh
Prime numbers between 1 and 50 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

11) Write a Shell program to find the sum of square of individual digits of a number.

program:

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

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./11.sh
Enter a number:
234
The sum of square of digits is: 29
```

12) Write a Shell program to count the number of vowels in a line of text.

program:

```
echo "Enter a string:"
read str
count=$(echo $str | grep -o -i "[aeiou]" | wc -l)
echo "Number of vowels: $count"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./12.sh
Enter a string:
sreyas
Number of vowels: 2
```

13) Write a Shell program to display student grades.

program:

```
calculate_grade() {
   if [ $1 -ge 90 ]; then
      grade="A"
   elif [ $1 -ge 80 ]; then
      grade="B"
   elif [ $1 -ge 70 ]; then
      grade="C"
   elif [ $1 -ge 60 ]; then
      grade="D"
   else
      grade="F"
   fi
   echo $grade
}

echo "Enter student name:"
read name
echo "Enter student's mark:"
read mark

grade=$(calculate_grade $mark)

echo "Student Name: $name"
echo "Student mark: $mark"
echo "Student Grade: $grade"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./13.sh
Enter student name:
sreyas
Enter student's mark:
90
Student Name: sreyas
Student mark: 90
Student Grade: A
```

14) Write a Shell program to find the smallest and largest numbers from a set of numbers.

program:

```
echo "Enter numbers separated by space:"
read -a numbers
largest=${numbers[0]}
smallest=${numbers[0]}
for num in "${numbers[@]}"; do
    if [ $num -gt $largest ]; then
        largest=$num
    fi
    if [ $num -lt $smallest ]; then
        smallest=$num
    fi
done
echo "Largest: $largest"
echo "Smallest: $smallest"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./14.sh
Enter numbers separated by space:
10 20 4 5 94 3 45 16 71 55 5 62
Largest: 94
Smallest: 3
```

15) Write a Shell program to find the smallest digit from a number.

program:

```
echo "Enter a number:"
read num
smallest=$(echo $num | grep -o "[0-9]" | sort | head -n1)
echo "Smallest digit: $smallest"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./15.sh
Enter a number:
2549671
Smallest digit: 1
```

16) 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.

program:

```
sum=0
for ((i=50; i<=100; i++)); do
   if [ $((i % 3)) -eq 0 ] && [ $((i % 5)) -ne 0 ]; then
      sum=$((sum + i))
   fi
done
echo "Sum of numbers divisible by 3 and not by 5 between 50 and 100: $sum"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./16.sh
Sum of numbers divisible by 3 and not by 5 between 50 and 100: 1050
```

17) Write a Shell program to find the second highest number from a set of numbers.

program:

```
echo "Enter numbers separated by space:"
read -a numbers
IFS=$'\n' sorted=($(sort -n <<<"${numbers[*]}"))
len=${#sorted[@]}
echo "Second highest number: ${sorted[len-2]}"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./17.sh
Enter numbers separated by space:
10 12 96 41 20 30 14 11 33
Second highest number: 41
```

18) Write a Shell program to find the sum of digits of a number using function.

program:

```
echo "Enter a number:"
read num
sum_digits() {
   local n=$1
   local sum=0
   while [ $n -gt 0 ]; do
      sum=$((sum + n % 10))
      n=$((n / 10))
   done
   echo $sum
}
echo "Sum of digits: $(sum_digits $num)"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./18.sh
Enter a number:
1256
Sum of digits: 14
```

19) Write a Shell program to print the reverse of a number using function.

program:

```
echo "Enter a number:"
read num
reverse() {
   local n=$1
   local rev=0
   while [ $n -gt 0 ]; do
      remainder=$((n % 10))
      rev=$((rev * 10 + remainder))
      n=$((n / 10))
   done
   echo $rev
}
echo "Reverse of $num is: $(reverse $num)"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./19.sh
Enter a number:
2546
Reverse of 2546 is: 6452
```

20) Write a Shell program to find the factorial of a number using for loop.

program:

```
echo "Enter a number:"
read num
fact=1
for ((i=1; i<=num; i++)); do
   fact=$((fact * i))
done
echo "Factorial of $num is: $fact"
```

output:

```
mca@mca-HP-Z238-Microtower-Workstation:~/sreyas/sem2/Networking/sh$ ./20.sh
Enter a number:
5
Factorial of 5 is: 120
```

21) Write a Shell program to generate Fibonacci series.

program:

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

output:

22) Write a shell script, which receives two filenames as arguments. It checks whether the two files contents are same or not. If they are same then second file is deleted.

program:

```
if [ $# -ne 2 ]; then
   echo "Usage: $0 <file1> <file2>"
   exit 1
fi
if cmp -s "$1" "$2"; then
   echo "Files are the same. Deleting $2"
   rm $2
else
   echo "Files are different"
fi
```

output:

23) Write a Menu driven Shell script that Lists current directory, Prints Working Directory, displays Date and displays Users logged in.

program:

```
PS3="Select option: "
select opt in "List current directory" "Print working directory" "Display date"
"Display users logged in" "Exit"; do
   case $opt in
      "List current directory")
         ls
         ;;
      "Print working directory")
         pwd
         ;;
      "Display date")
         date
         ;;
      "Display users logged in")
         who
         ;;
      "Exit")
         break
         ;;
      *)
         echo "Invalid option"
         ;;
   esac
done
```

output:

24) Shell script to check executable rights for all files in the current directory, if a file does not have the execute permission then make it executable.

program:

```
for file in *; do
   if [ -f $file ] && [ ! -x $file ]; then
      chmod +x $file
      echo "$file is made executable"
   fi
done
```

output:

25) Write a Shell program to generate all combinations of 1, 2, and 3 using loop.

program:

```
for i in 1 2 3; do
   for j in 1 2 3; do
      for k in 1 2 3; do
         echo "$i$j$k"
      done
   done
done
```

output:

26) Write a Shell program to create the number series.

program:

```
echo "Enter the number of terms:"
read n
echo "Number series:"
for ((i=1; i<=n; i++)); do
   echo -n "$i "
done
```

output:

27) Write a Shell program to create Pascal's triangle.

program:

```
echo "Enter the number of rows:"
read rows
for ((i=0; i<rows; i++)); do
   for ((j=0; j<=i; j++)); do
      if [ $j -eq 0 ] || [ $i -eq $j ]; then
         coef=1
      else
         num=$((i-j+1))
         den=$j
         coef=$((coef * num / den))
      fi
      echo -n "$coef "
   done
   echo
done
```

output:

28) Write a Decimal to Binary Conversion Shell Script.

program:

```
echo "Enter a decimal number:"
read decimal
echo "Binary conversion: $(echo "obase=2; $decimal" | bc)"
```

output:

29) Write a Shell Script to Check Whether a String is Palindrome or not.

program:

```
echo "Enter a string:"
read str
reverse=$(echo $str | rev)
if [ "$str" = "$reverse" ]; then
   echo "$str is a palindrome"
else
   echo "$str is not a palindrome"
fi
```

output:

30) Write a shell script to find out the unique words in a file and also count the occurrence of each of these words.

program:

```
echo "Enter file name:"
read filename
awk '{for(i=1;i<=NF;i++) a[$i]++} END {for(k in a) print k, a[k]}' $filename
```

output:

31) Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

program:

```
echo "Enter directory path:"
read dir
grep -roh "Linux" $dir | wc -w
```

output:

32) Write a shell script to validate password strength. Here are a few assumptions for the password string. (Length – minimum of 8 characters. Contain both alphabet and number. Include both the small and capital case letters.)

program:

```
echo "Enter password:"
read password
if [[ ${#password} -lt 8 ]]; then
   echo "Password length should be at least 8 characters"
   exit 1
fi
if ! [[ $password =~ [0-9] ]]; then
   echo "Password should contain at least one digit"
   exit 1
fi
if ! [[ $password =~ [A-Z] ]]; then
   echo "Password should contain at least one uppercase letter"
   exit 1
fi
if ! [[ $password =~ [a-z] ]]; then
   echo "Password should contain at least one lowercase letter"
   exit 1
fi
echo "Password is strong"
```

output:

33) Write a shell script to print the count of files and subdirectories in the specified directory.

program:

```
echo "Enter directory path:"
read dir
echo "Number of files and subdirectories: $(find $dir -type d -or -type f | wc -l)"
```

output:

34) Write a shell script to reverse the list of strings and reverse each string further in the list.

program:

```
echo "Enter strings separated by space:"
read -a strings
for ((i=0; i<${#strings[@]}; i++)); do
   rev=$(echo ${strings[i]} | rev)
   reversed_strings[$i]=$rev
done
echo "Reversed list of strings:"
for string in "${reversed_strings[@]}"; do
   echo "$(echo $string | rev)"
done
```

output: