Output :-

enter first no. :
4
Enter second no. :
6.
1. Addition
2. Subtraction
3. Multiplication
4. division
1.
Sum 10

Output :-

Enter the number
6
Friday
Enter the number
8
Enter value b/w 1 to 7

POORNIMA

Experiment - 5

1. Write a shell script to make a basic calculator which performs addition, substraction, multiplication & division.

```
echo "enter first no."
read a
echo "enter second no."
read b
echo "1 Addition"
echo "2 Substraction"
echo "3 Multiplication"
echo "4 division"
read c
case $c in
1) echo "Sum $((a+b))";;
2) echo "Substraction $((a-b))";;
3) echo "Multiplication $((a*b))";;
4) echo "division $((a/b))";;
*) echo "enter valid operation"
esac.
```

2. Write shell script to print days a week.

```
echo "Enter the no."
read n
case $n in
1) echo "Sunday";;
2) echo "Monday";;
```

Output :-

Jan - 31 day
Mar - 31 days
May - 31 days
Jul - 31 days

```
3-) echo "Tuesday" ;;
4.) echo "Wednesday" ;;
5.) echo "Thursday" ;;
6.) echo "Friday" ;;
7) echo "Saturday" ;;
*) echo "enter a valid no. b/w 1 to 7" ;;
   esac
```

3. Aim:- To print 4 month of 31 days

```
declare -i c=1
For m in(1.....12);
do
a=$(date -d "$m/1 +1 month -1 day" ." +%b-
    % d days " ;)
echo "$a"
done.
```

Bhuvana
11/8/24

Output :-

Enter the number : 153
Armstrong no.

Enter the number : 123
Not an armstrong no.

POORNIMA

Experiment - 06

1. Aim :- Write a shell script to find a no. is Armstrong or not.

```
echo "Enter the number"
read n
Function ams
{
    t = $n
    s = 0
    b = 0
    c = 10
while [$n -gt $b]
    do
    r = $((n%c))
    i = $((r*r*r))
    s = $((s+i))
    n = $((n/c))
    done
    echo $s
if [$s == $t]
then
    echo "Armstrong number"
    else
echo "Not an Armstrong number"
    Fi
}
result = "ams $n"
    echo "$ result"
```

Output :-

Enter the no. : 16461
palidrome

Enter the no. : 14
Not a palidrome.

POORNIMA

2. Aim: Write a shell script to find a no. is palindrome or not

```
echo "Enter the no."
read n
Functional pal
{
number = $n
reverse = 0
while [$n -gt 0]
ab
a = `expr $ n % 10`
n = `expr $ n / 10`
reverse = `expr $ reverse \* 10 + $ a`
done
echo $ reverse
IF [$ number - eq $ reverse]
then
echo "Number is palindrome"
else
echo "no. is not palindrome"
Fi
}
a = "pal $n"
echo "$a"
```

POORNIMA

2. Aim : Write a shell script to find a no. is palindrome or not

```
echo "Enter the no."
read n
Functional pal
{
number = $n
reverse = 0
while [$n -gt 0]
ab
a = `expr $ n % 10`
n = `expr $ n / 10`
reverse = `expr $ reverse \* 10 + $ a`
done
echo $ reverse
IF [$ number - eq $ reverse]
then
echo "Number is palindrome"
else
echo "no. is not palindrome"
Fi
}
r = "pal $n"
echo "$r"
```

Output :-

How many no. of terms to be generated ?
5

Fiboracci series : 0, 1, 2, 3, 5

POORNIMA

3. Write a shell script to print Fiboracci series e

```
echo "How many no. of terms to generated"
    read n
    Function Fib
{
    n=0
    y=1
    i=2
echo "Fibonacci series up to $ n terms "
        echo "$n"
        echo "$y"
while [$i -lt $n]
do
i="expr $i + 1"
z=" expr $x + $y"
    echo "$z"
        n=$y
        y=$z
        done
}
g=' fib $n'
    echo "$g"
```

Output :-

Number : 853
The number is prime

Output :-

Enter the number : 4
Binary equivalent : 100

POORNIMA

4. Aim:- Write a shell script to find prime number

```
number = 53
i = 2
Flag = 0
while test $i -le `expr $number 12`
do
if test `expr $number % $i` -eq 0
  then
  Flag = 1
  Fi
  i = `expr $i + 1`
done if test $Flag -eq 1
  then
echo " The number is not prime
else
echo " The no. is prime"
Fi
```

5. Aim:- Write a shell script to convert binary to decimal & decimal to binary.

```
echo " Enter the number"
  read n
  val = 0
  power = 1
while [$n -neo]
do
r = " expr $n % 2"
```

Page No.....

POORNIMA

```
val = " expr $r1 * $ power + $ val "
power = " expr $ power \* 16 '
n = "expr $n \ 12"
done
echo " Binary equivalent - $ val "
```

13/6/24

Output :-

Please enter the no. of lines : 4
The pattern is :

```
* * * *
* * * *
* * * *
* * * *
```

Output :-

Please enter the no. of lines = 4
The pattern is

```
* * * *
*
*
*
```

POORNIMA

Experiment - 7

1. Aim :- To print square pattern

```
echo -n "please enter the no. of lines"
    read lines
    echo " The pattern is :"
For (( row = 1 ; row <= lines ; ++ row ))
do
For (( column = 1 ; column <= lines ; ++ column ))
  do
echo -n "*"
    done
    echo
    done
```

2. Aim :- To print hollow square pattern

```
echo -n "please enter to no. of lines:"
    read lines
    echo " The pattern is :"
For (( row = 1 ; row <= lines ; ++ row ))
  do
For (( col = 1 ; col <= lines ; ++ col ))
  do
iF (( $ row = 1 || $ row = $ lines || $ col = $ lines ))
    then
echo -n "*"
```

Output :-

Please enter the no. lines : 4

The patteern is

```
* * * * * *
* * * * * *
* * * * * *
* * * * * *
```

Output :-

Enter the no. of line : 4

The patteern is

```
   *
  * *
 * * *
* * * *
```

```
else
echo -n "
Fi
done
echo
done
```

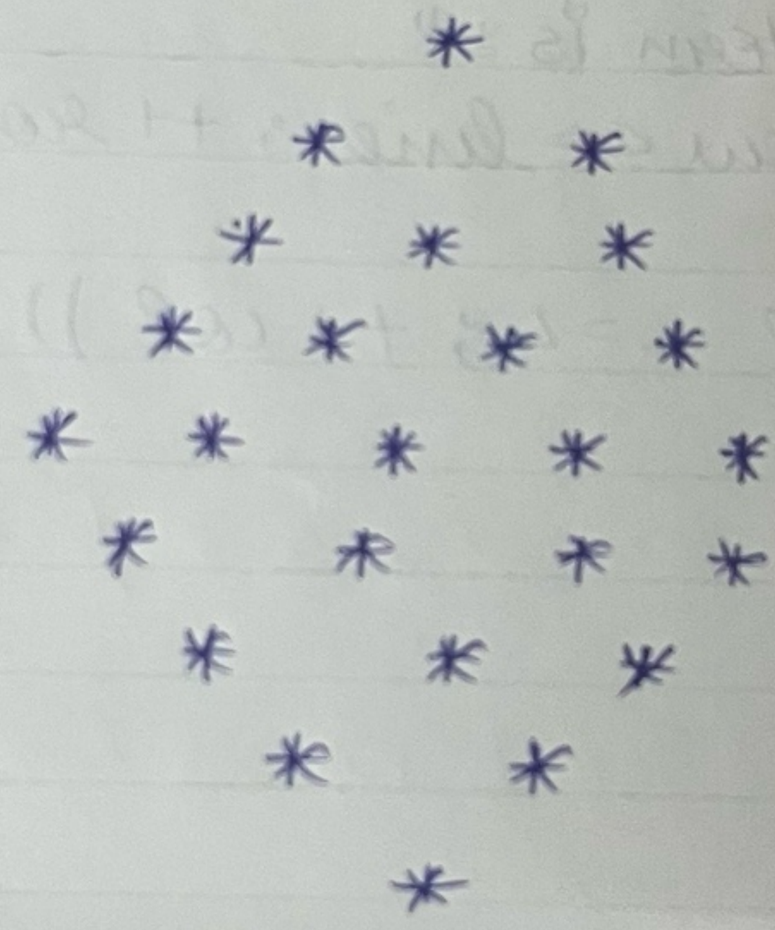
3. Aim:- To print a rectangular pattern

```
echo -n "please enter the no. of lines :"
read lines
echo "The pattern is :"
For ((row = 1; row <= lines ; ++row))
do
For ((col = 1 ; col <= 4; ++col ))
do
echo -n "*"
done
echo
done.
```

4. Aim :- To print triangle pattern.

```
echo -n "Enter the no. of lines:"
read lines
For((row=1 ; row <= lines; row++))
do
For((col = row ; col <= lines ; col ++))
echo -n " ";
```

Output :-



done
For((n=1 ; n<= row ; n++))
do
echo -ne "*" ;
For((i=1 ; i<= row ; i++))
do
echo -ne "*" ;
done
echo
done

5. Aim :- To print diamond pattern

echo "Enter value of n:"
read num
For((i=1 ; i<= $num ; i++))
do
For((j= $ ~~space~~ num ; j>=i ; j--))
do
echo -n " "
do
For((c=1 ; ; c<=i ; c++))
do
echo -n "*"
sum = "expr $sum +1"
done
do
echo -n "*"

POORNIMA

```
sum = "espr $sum +1"
done
echo " "
done
```

Output
17/5/24

Output :-

array size = 4
element = {12, -1, 0, 8}

O/P
= 19

enter element to search : 12
12 is found at position 1

POORNIMA

Experiment - 8

1. Aim:- Write C program to read, print, sum, reverse search of sort array element in ascending or decending order.

```
#include <stdio.h>
    int main ()
{
  int arr[100], size, i, sum = 0;
printf("Enter array size /n");
printf("Enter the array element /n");
    for (i=0; i< size ;i++);
{
  scanf("%d", &arr [i]); }
  For(i=0; i < size ; i++) {
      sum = sum + arr [i]
printf("sum of array %d /n", sum);
    return 0;
printf("\n Enter the element to search:");
    int to search;
  scanf("%d ", & to search);
  Found 0;
  For (i=0 ; i< size ; i++) {
  iF ( arr [i] = to search) {
      found = 1;
      break }}
  iF (found == 1) {
```

```
printf("\n%d is found at positive %d", to search
                    , i+1); }
        else
{
    print("%d is not found in line array :
                    to search);
    }
    return 0;
    }
```

sorting

```
# include <stdio.h>
void swap (int * xp, int * yp) {
            int temp = xp;
            * xp  = * xy;
            * xy  = temp ; }
void selection sort (int arr[], int n) {
            int i, j, min - idx;
For (i = 0; i < n - 1; i++)
    { min - idx = i;
For (j = i + 1; j < n ; j++) {
    if (arr [j] < arr [min - idx])
            min - idx = j ;
swap (arr [min - idx], &arr [i];
 } }
void print array (int arr[], int size)
{
int i ;
for (i > 0; i < &in ; i++) {
```

Output :-

Original array

0, 23, 14, 12, 9

Sorted array in asending order;

0, 9, 12, 14, 23

POORNIMA

POORNIMA

```
    printf("%d", arr[i]);
    printf("\n"); }
        int main()
{
int arr[] = [0,23,14,12,9];
int n = size of (array) / size of (arr[0])
  printf("original array : \n");
   print array (arr, n);
   selectionsort (arr, n);
printf("\n sort array in ascending order");
      printf [Array (arr, n);
          return 0;
}
```

Output

17/5/24