AIM:-

Write a shell script to implement a simple calculator (using 'if') and using switch case.

- ## **If-elif-fi**

The if...elif...fi statement is the one level advance form of control statement that allows Shell to make correct decision out of several conditions. This code is just a series of *if* statements, where each *if* is part of the *else* clause of the previous statement. Here statement(s) are executed based on the true condition, if none of the condition is true then *else* block is executed.

Syntax:

```
if [ expression 1 ]
then
   Statement(s) to be executed if expression 1 is true
elif [ expression 2 ]
then
   Statement(s) to be executed if expression 2 is true
elif [ expression 3 ]
then
   Statement(s) to be executed if expression 3 is true
else
   Statement(s) to be executed if no expression is true
fi
```

- **case-esac**

The basic syntax of the **case...esac** statement is to give an expression to evaluate and to execute several different statements based on the value of the expression.

The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

Syntax:

```
case word in
  pattern1)
    Statement(s) to be executed if pattern1 matches
    ;;
  pattern2)
    Statement(s) to be executed if pattern2 matches
    ;;
  pattern3)
    Statement(s) to be executed if pattern3 matches
    ;;
  *)
    Default condition to be executed
    ;;
esac
```

- **while loop**

The **while** loop enables you to execute a set of commands repeatedly until some condition occurs. It is usually used when you need to manipulate the value of a variable repeatedly.
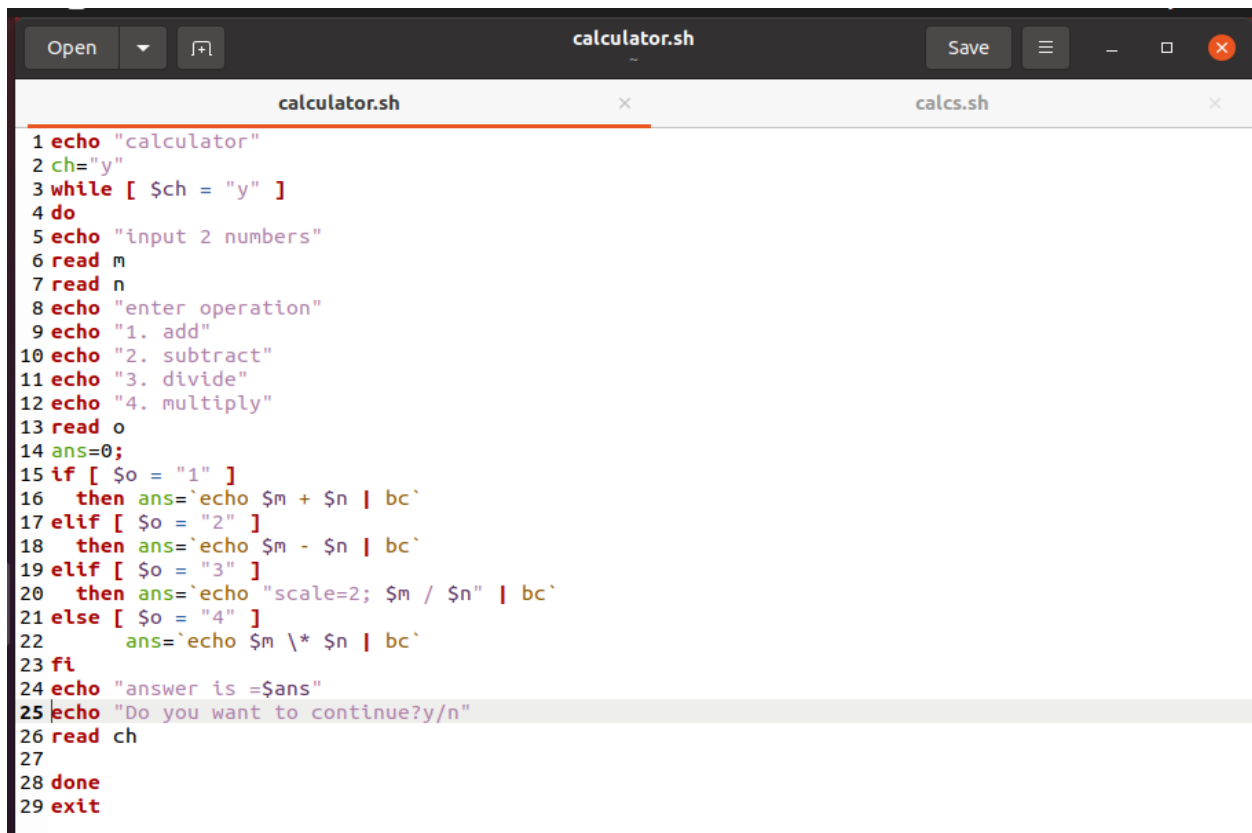
Syntax:

while command

do

  Statement(s) to be executed if command is true

done

OUTPUT


```
1 echo "calculator"
2 ch="y"
3 while [ $ch = "y" ]
4 do
5 echo "input 2 numbers"
6 read m
7 read n
8 echo "enter operation"
9 echo "1. add"
10 echo "2. subtract"
11 echo "3. divide"
12 echo "4. multiply"
13 read o
14 ans=0;
15 if [ $o = "1" ]
16    then ans=`echo $m + $n | bc`
17 elif [ $o = "2" ]
18    then ans=`echo $m - $n | bc`
19 elif [ $o = "3" ]
20    then ans=`echo "scale=2; $m / $n" | bc`
21 else [ $o = "4" ]
22        ans=`echo $m \* $n | bc`
23 fi
24 echo "answer is =$ans"
25 echo "Do you want to continue?y/n"
26 read ch
27
28 done
29 exit
```

```
anagha@anagha-da:~$ bash calculator.sh
calculator
input 2 numbers
4
5
enter operation
1. add
2. subtract
3. divide
4. multiply
1
answer is =9
Do you want to continue?y/n
y
input 2 numbers
5
3
enter operation
1. add
2. subtract
3. divide
4. multiply
2
answer is =2
Do you want to continue?y/n
y
input 2 numbers
5.1
2.5
enter operation
1. add
2. subtract
3. divide
4. multiply
3
answer is =2.04
Do you want to continue?y/n
y
input 2 numbers
8
2.5
enter operation
1. add
2. subtract
3. divide
4. multiply
4
answer is =20.0
Do you want to continue?y/n
n
anagha@anagha-da:~$
```
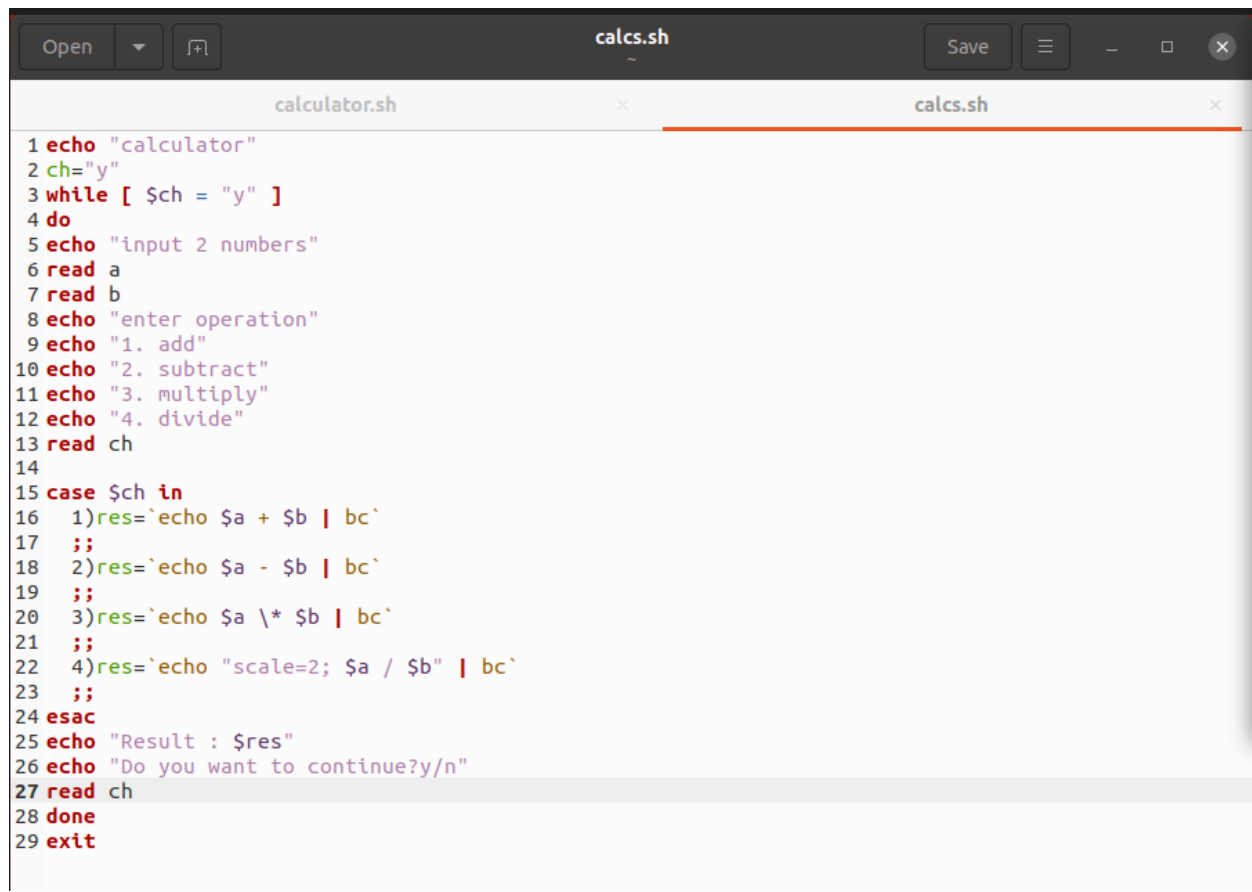
Open ▼ 🔲

Save ☰ — ▢ ✕

```
 1 echo "calculator"
 2 ch="y"
 3 while [ $ch = "y" ]
 4 do
 5 echo "input 2 numbers"
 6 read a
 7 read b
 8 echo "enter operation"
 9 echo "1. add"
10 echo "2. subtract"
11 echo "3. multiply"
12 echo "4. divide"
13 read ch
14
15 case $ch in
16   1)res=`echo $a + $b | bc`
17   ;;
18   2)res=`echo $a - $b | bc`
19   ;;
20   3)res=`echo $a \* $b | bc`
21   ;;
22   4)res=`echo "scale=2; $a / $b" | bc`
23   ;;
24 esac
25 echo "Result : $res"
26 echo "Do you want to continue?y/n"
27 read ch
28 done
29 exit
```

```
anagha@anagha-da:~$ bash calcs.sh
calculator
input 2 numbers
5.1
4.2
enter operation
1. add
2. subtract
3. multiply
4. divide
1
Result : 9.3
Do you want to continue?y/n
y
input 2 numbers
6.1
5.1
enter operation
1. add
2. subtract
3. multiply
4. divide
2
Result : 1.0
Do you want to continue?y/n
y
input 2 numbers
8.2
2.0
enter operation
1. add
2. subtract
3. multiply
4. divide
3
Result : 16.4
Do you want to continue?y/n
y
input 2 numbers
9.3
3.1
enter operation
1. add
2. subtract
3. multiply
4. divide
4
Result : 3.00
Do you want to continue?y/n
n
anagha@anagha-da:~$
```