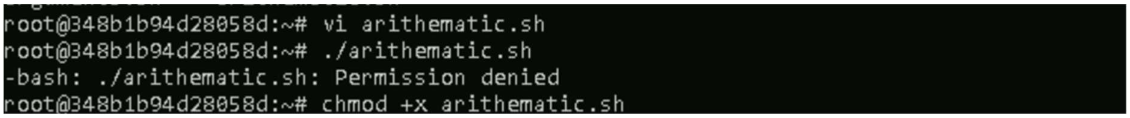
1. Arithmetic Operation

Step 1:

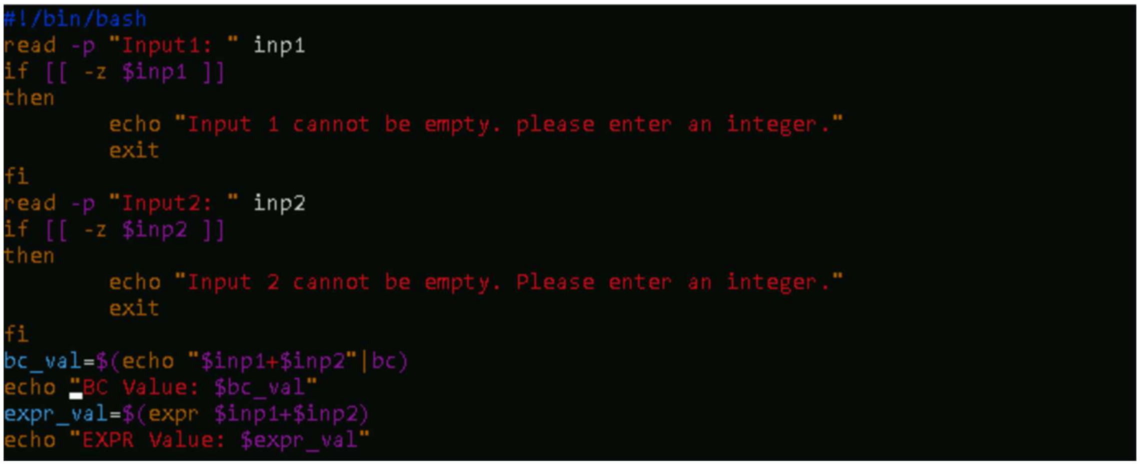
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~# vi arithematic.sh
root@348b1b94d28058d:~# ./arithematic.sh
-bash: ./arithematic.sh: Permission denied
root@348b1b94d28058d:~# chmod +x arithematic.sh
```

Step 2:

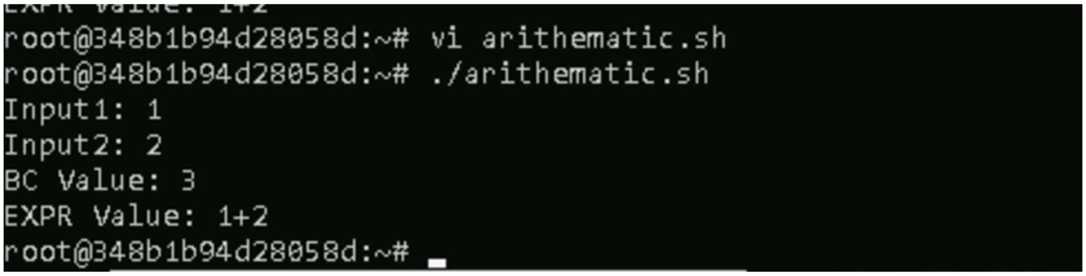
Write script in the file



```
#!/bin/bash
read -p "Input1: " inp1
if [[ -z $inp1 ]]
then
        echo "Input 1 cannot be empty. please enter an integer."
        exit
fi
read -p "Input2: " inp2
if [[ -z $inp2 ]]
then
        echo "Input 2 cannot be empty. Please enter an integer."
        exit
fi
bc_val=$(echo "$inp1+$inp2"|bc)
echo "BC Value: $bc_val"
expr_val=$(expr $inp1+$inp2)
echo "EXPR Value: $expr_val"
```

Step 3:

Run the script and get the output.

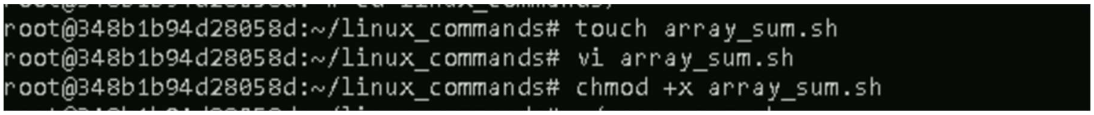


```
EXPR Value: 1+2
root@348b1b94d28058d:~# vi arithematic.sh
root@348b1b94d28058d:~# ./arithematic.sh
Input1: 1
Input2: 2
BC Value: 3
EXPR Value: 1+2
root@348b1b94d28058d:~#
```

2. Sum of positive numbers

Step 1:

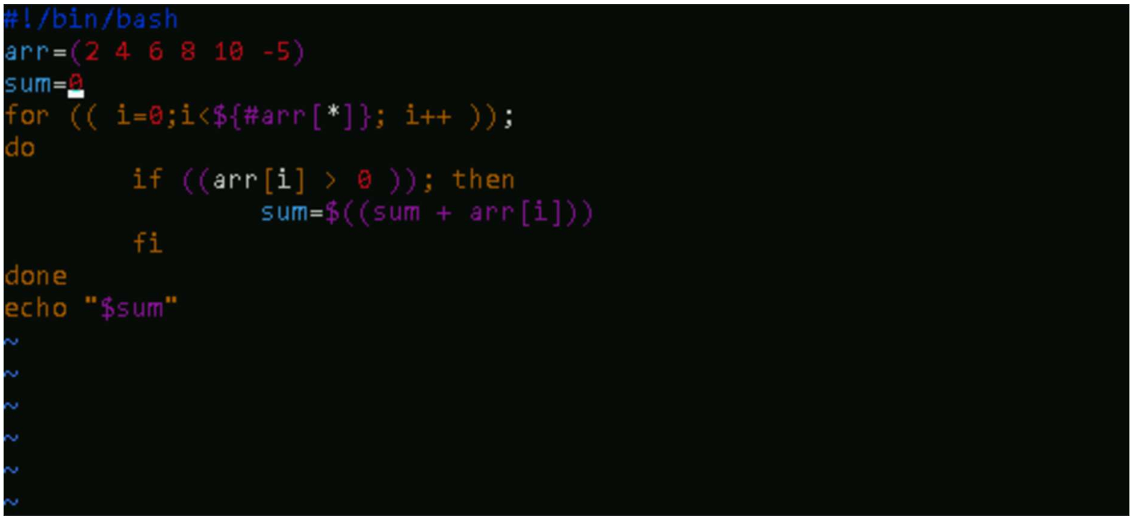
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch array_sum.sh
root@348b1b94d28058d:~/linux_commands# vi array_sum.sh
root@348b1b94d28058d:~/linux_commands# chmod +x array_sum.sh
```

Step 2:

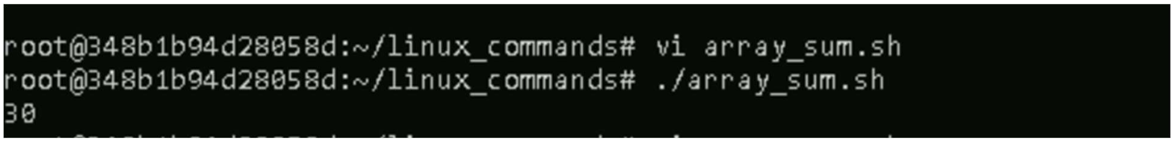
Write script in the file.



```
#!/bin/bash
arr=(2 4 6 8 10 -5)
sum=0
for (( i=0;i<${#arr[*]}; i++ ));
do
        if ((arr[i] > 0 )); then
                sum=$((sum + arr[i]))
        fi
done
echo "$sum"
~
~
~
~
~
~
```

Step 3:

Run the script and get the output.

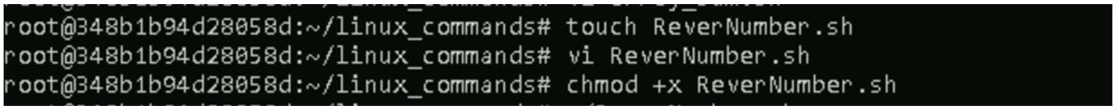


```
root@348b1b94d28058d:~/linux_commands# vi array_sum.sh
root@348b1b94d28058d:~/linux_commands# ./array_sum.sh
30
```

3. Reverse a number

Step 1:

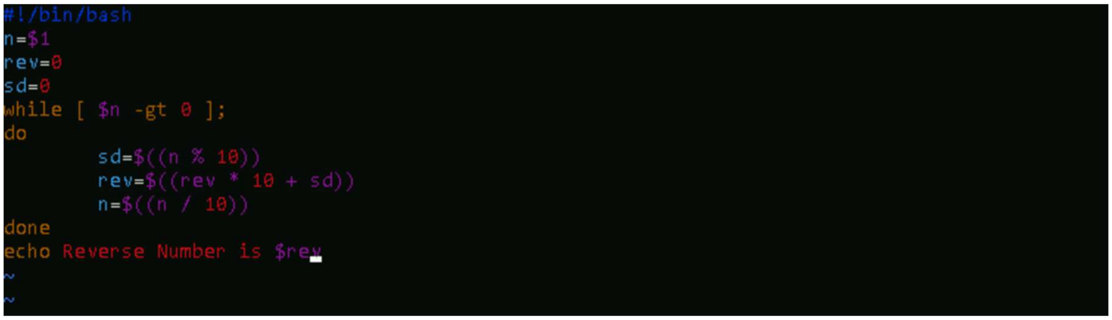
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch ReverNumber.sh
root@348b1b94d28058d:~/linux_commands# vi ReverNumber.sh
root@348b1b94d28058d:~/linux_commands# chmod +x ReverNumber.sh
```

Step 2:

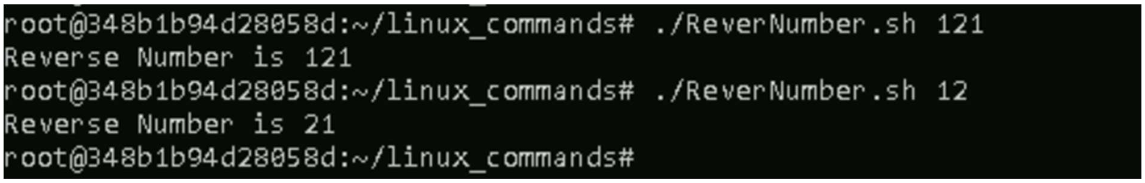
Write script in the file.



```
#!/bin/bash
n=$1
rev=0
sd=0
while [ $n -gt 0 ];
do
        sd=$((n % 10))
        rev=$((rev * 10 + sd))
        n=$((n / 10))
done
echo Reverse Number is $rev
~
~
```

Step 3:

Run the script and get the output.

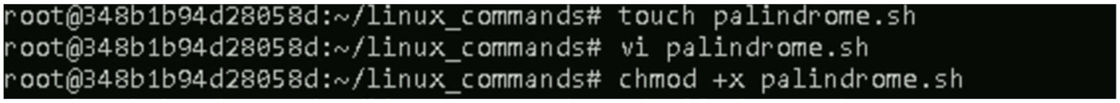


```
root@348b1b94d28058d:~/linux_commands# ./ReverNumber.sh 121
Reverse Number is 121
root@348b1b94d28058d:~/linux_commands# ./ReverNumber.sh 12
Reverse Number is 21
root@348b1b94d28058d:~/linux_commands#
```

4. Palindrome

Step 1:

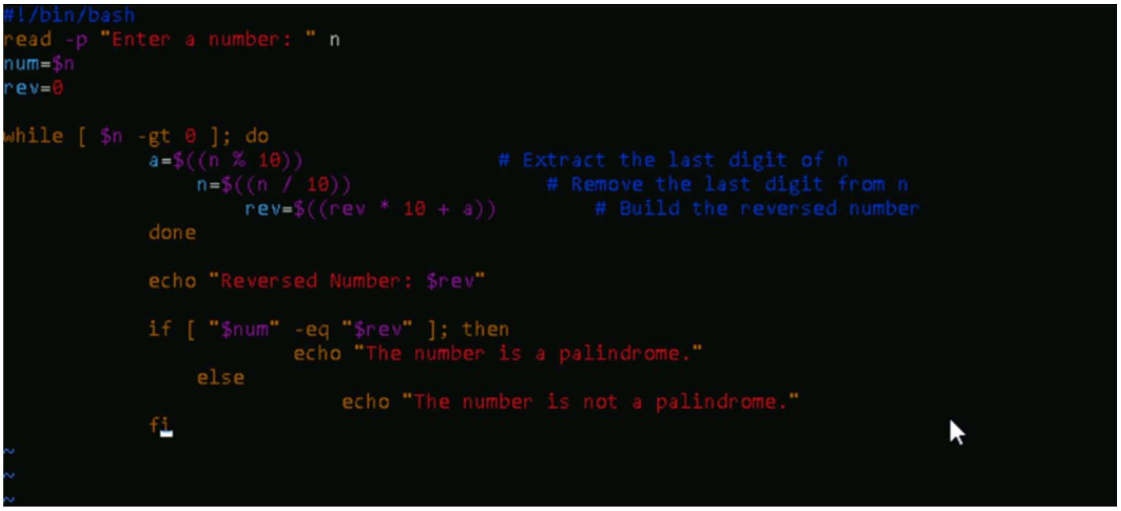
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch palindrome.sh
root@348b1b94d28058d:~/linux_commands# vi palindrome.sh
root@348b1b94d28058d:~/linux_commands# chmod +x palindrome.sh
```

Step 2:

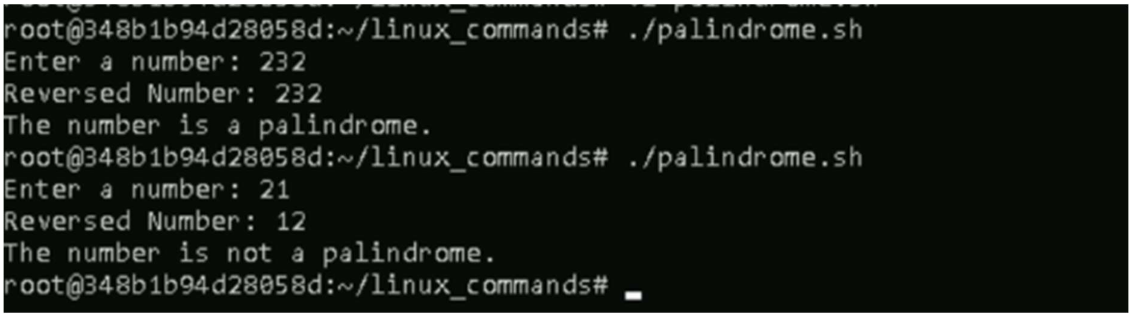
Write script in the file.



```
#!/bin/bash
read -p "Enter a number: " n
num=$n
rev=0

while [ $n -gt 0 ]; do
            a=$((n % 10))                # Extract the last digit of n
                n=$((n / 10))                # Remove the last digit from n
                    rev=$((rev * 10 + a))        # Build the reversed number
            done

            echo "Reversed Number: $rev"

            if [ "$num" -eq "$rev" ]; then
                        echo "The number is a palindrome."
                else
                            echo "The number is not a palindrome."
            fi
~
~
~
```

Step 3:

Run the script and get the output.

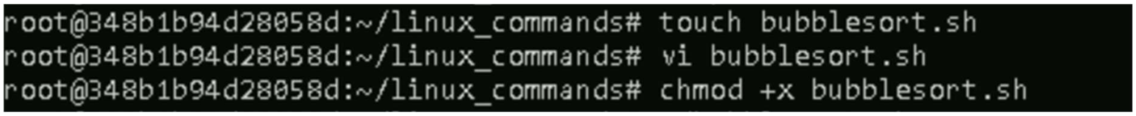


```
root@348b1b94d28058d:~/linux_commands# ./palindrome.sh
Enter a number: 232
Reversed Number: 232
The number is a palindrome.
root@348b1b94d28058d:~/linux_commands# ./palindrome.sh
Enter a number: 21
Reversed Number: 12
The number is not a palindrome.
root@348b1b94d28058d:~/linux_commands#
```

5. Bubble sort

Step 1:

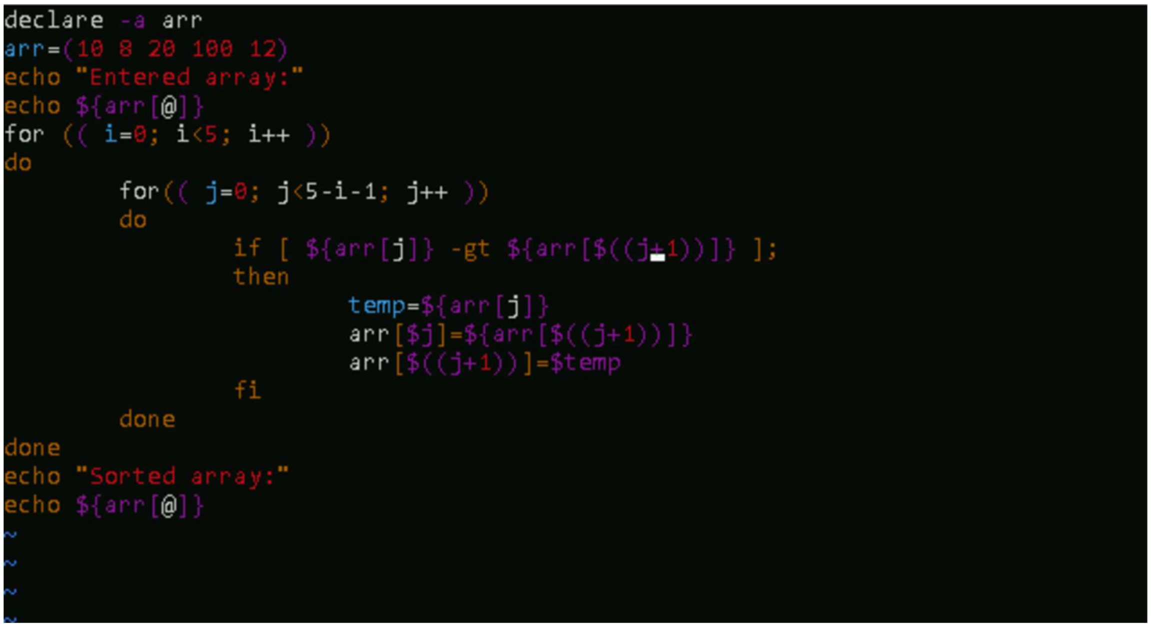
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch bubblesort.sh
root@348b1b94d28058d:~/linux_commands# vi bubblesort.sh
root@348b1b94d28058d:~/linux_commands# chmod +x bubblesort.sh
```

Step 2:

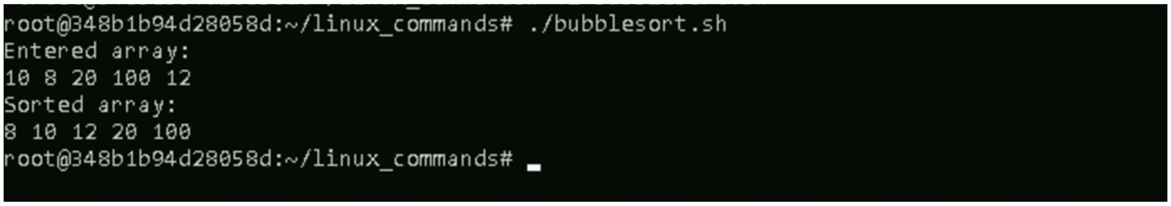
Write script in the file.



```
declare -a arr
arr=(10 8 20 100 12)
echo "Entered array:"
echo ${arr[@]}
for (( i=0; i<5; i++ ))
do
        for(( j=0; j<5-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:"
echo ${arr[@]}
~
~
~
~
```

Step 3:

Run the script and get the output.

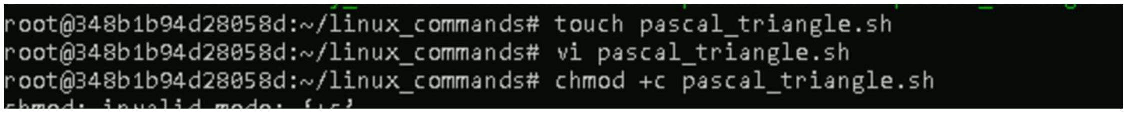


```
root@348b1b94d28058d:~/linux_commands# ./bubblesort.sh
Entered array:
10 8 20 100 12
Sorted array:
8 10 12 20 100
root@348b1b94d28058d:~/linux_commands#
```

6. Pascal Triangle

Step 1:

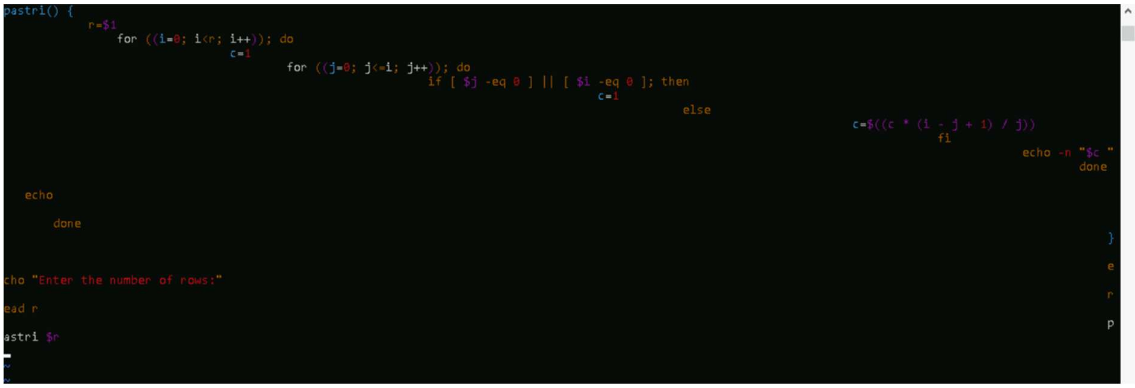
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch pascal_triangle.sh
root@348b1b94d28058d:~/linux_commands# vi pascal_triangle.sh
root@348b1b94d28058d:~/linux_commands# chmod +c pascal_triangle.sh
chmod: invalid mode: '+c'
```

Step 2:

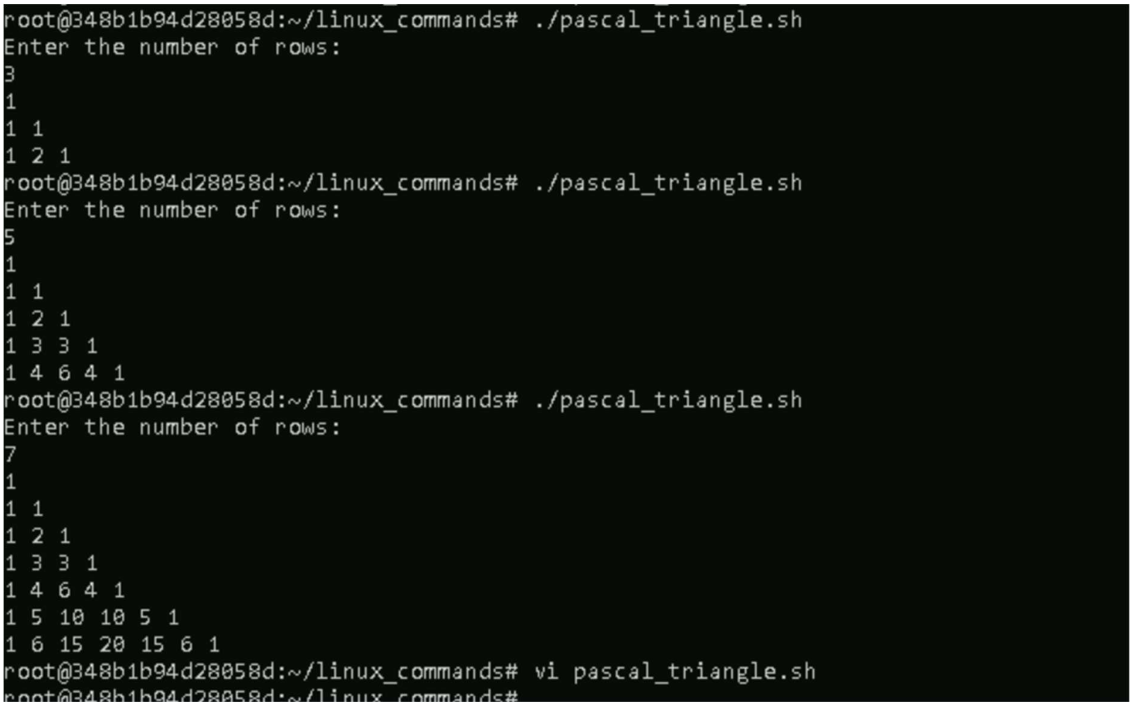
Write script in the file.



```
pastri() {
            r=$1
                for ((i=0; i<r; i++)); do
                                c=1
                                        for ((j=0; j<=i; j++)); do
                                                            if [ $j -eq 0 ] || [ $i -eq 0 ]; then
                                                                                    c=1
                                                                                                else
                                                                                                                        c=$((c * (i - j + 1) / j))
                                                                                                                                    fi
                                                                                                                                                echo -n "$c "
                                                                                                                                                        done

   echo

       done

                                                                                                                                                            }

                                                                                                                                                            e
cho "Enter the number of rows:"
                                                                                                                                                            r
ead r
                                                                                                                                                            p
astri $r
~
~
```

Step 3:

Run the script and get the output.

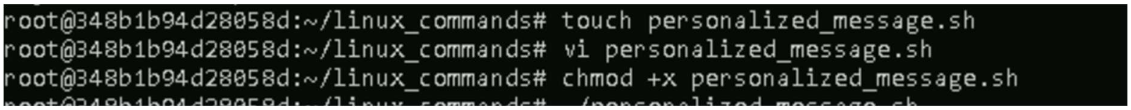


```
root@348b1b94d28058d:~/linux_commands# ./pascal_triangle.sh
Enter the number of rows:
3
1
1 1
1 2 1
root@348b1b94d28058d:~/linux_commands# ./pascal_triangle.sh
Enter the number of rows:
5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
root@348b1b94d28058d:~/linux_commands# ./pascal_triangle.sh
Enter the number of rows:
7
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
root@348b1b94d28058d:~/linux_commands# vi pascal_triangle.sh
root@348b1b94d28058d:~/linux_commands#
```

7. Personalized message

Step 1:

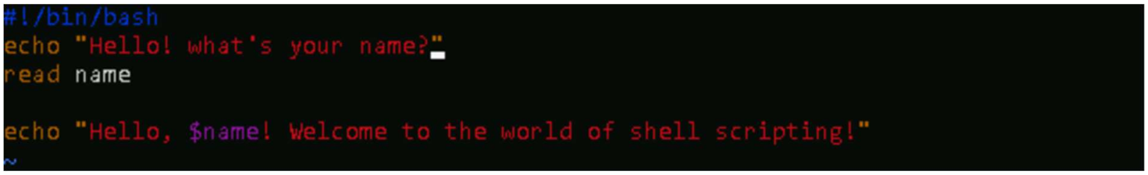
Create a shell script and open in vi editor



```
root@348b1b94d28058d:~/linux_commands# touch personalized_message.sh
root@348b1b94d28058d:~/linux_commands# vi personalized_message.sh
root@348b1b94d28058d:~/linux_commands# chmod +x personalized_message.sh
root@348b1b94d28058d:~/linux_commands# ./personalized_message.sh
```

Step 2:

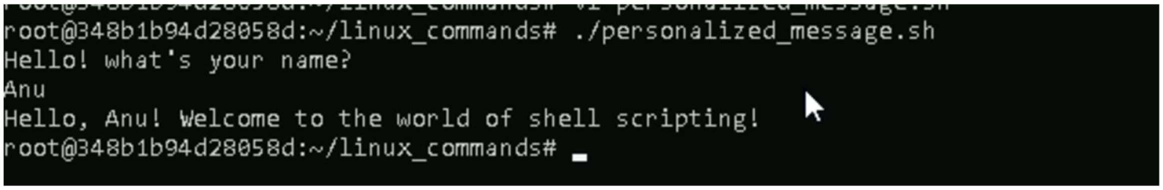
Write script in the file.



```
#!/bin/bash
echo "Hello! what's your name?"
read name

echo "Hello, $name! Welcome to the world of shell scripting!"
~
```

Step 3:

Run the script and get the output.



```
root@348b1b94d28058d:~/linux_commands# vi personalized_message.sh
root@348b1b94d28058d:~/linux_commands# ./personalized_message.sh
Hello! what's your name?
Anu
Hello, Anu! Welcome to the world of shell scripting!
root@348b1b94d28058d:~/linux_commands#
```