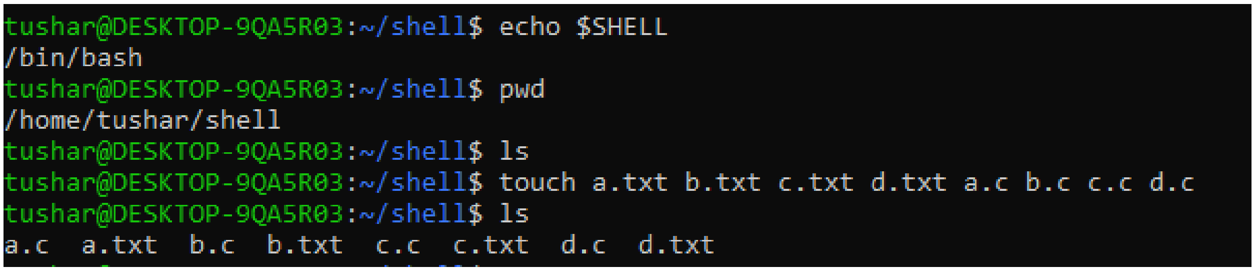
1. Display your current shell on terminal using **echo** command.
2. Display files and directories in present working directory using **echo** command.
3. create 4 files with **.txt** extension and 4 files with **.c** extension - display only **.txt** extension files using **echo** command in present working directory.

   - display only **.c** extension files using **echo** command in present working directory.(Hint: use **touch** command to create files)
4. Display all environment variables on terminal using command and verify few of them by printing them using **echo** command. (Hint: use **printenv** command)

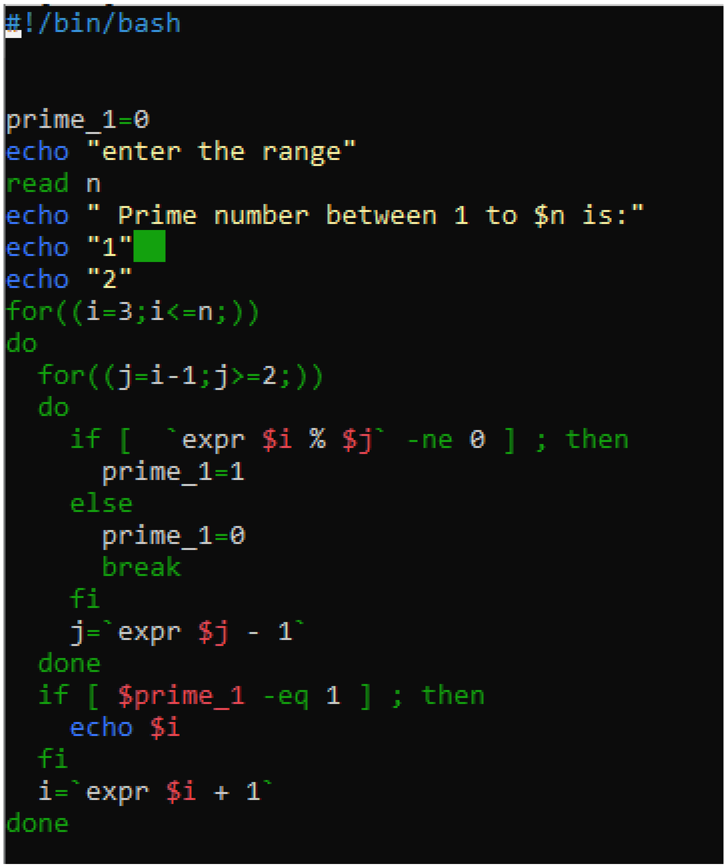


```
tushar@DESKTOP-9QA5R03:~/shell$ echo $SHELL
/bin/bash
tushar@DESKTOP-9QA5R03:~/shell$ pwd
/home/tushar/shell
tushar@DESKTOP-9QA5R03:~/shell$ ls
tushar@DESKTOP-9QA5R03:~/shell$ touch a.txt b.txt c.txt d.txt a.c b.c c.c d.c
tushar@DESKTOP-9QA5R03:~/shell$ ls
a.c  a.txt  b.c  b.txt  c.c  c.txt  d.c  d.txt
```

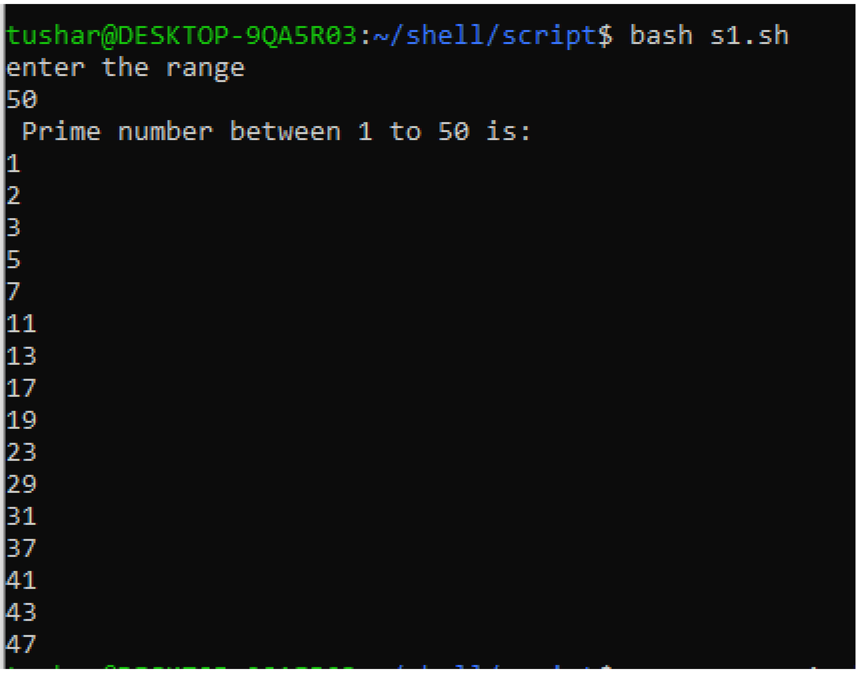
Part 1 (Shell scripting)

Note1: For the following assignments, accept only integer values from user.If user enters value other than integer display error message and terminate the program.
Note2: If you want to perform the following arithmetic operations with Floating point values make use of **Basic calculator** in your script.
(**Hint:** use **bc** command for Basic calculator)

1. Write a shell script to Print prime numbers from 1 to **n**. Read the value of **n** from user.

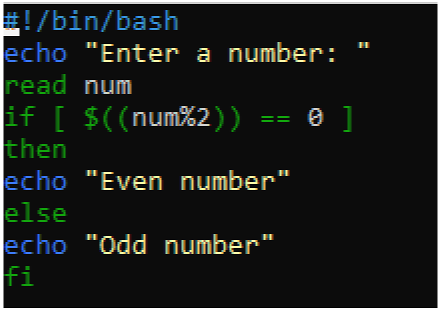


```
#!/bin/bash


prime_1=0
echo "enter the range"
read n
echo " Prime number between 1 to $n is:"
echo "1"
echo "2"
for((i=3;i<=n;))
do
  for((j=i-1;j>=2;))
  do
    if [  `expr $i % $j` -ne 0 ] ; then
      prime_1=1
    else
      prime_1=0
      break
    fi
    j=`expr $j - 1`
  done
  if [ $prime_1 -eq 1 ] ; then
    echo $i
  fi
  i=`expr $i + 1`
done
```

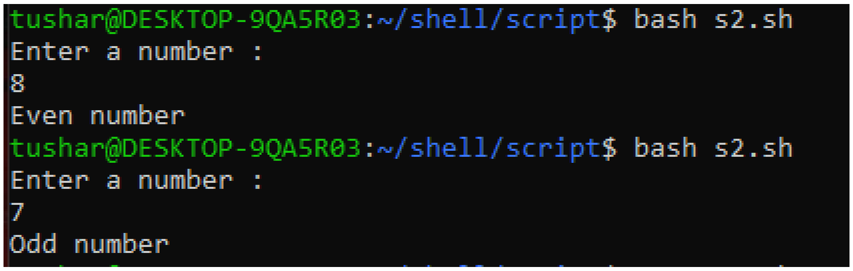


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s1.sh
enter the range
50
 Prime number between 1 to 50 is:
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

2. Write a script to check given number is even or odd.

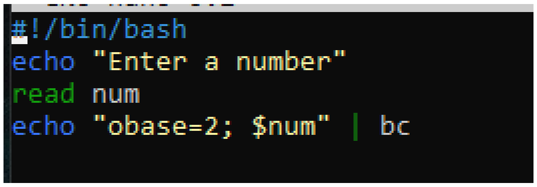


```
#!/bin/bash
echo "Enter a number: "
read num
if [ $((num%2)) == 0 ]
then
echo "Even number"
else
echo "Odd number"
fi
```

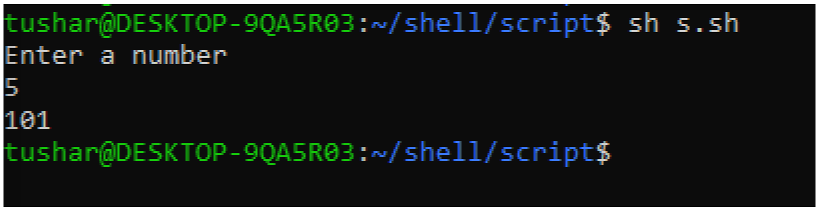


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s2.sh
Enter a number :
8
Even number
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s2.sh
Enter a number :
7
Odd number
```

3. Write a shell script to convert a decimal number to binary number.

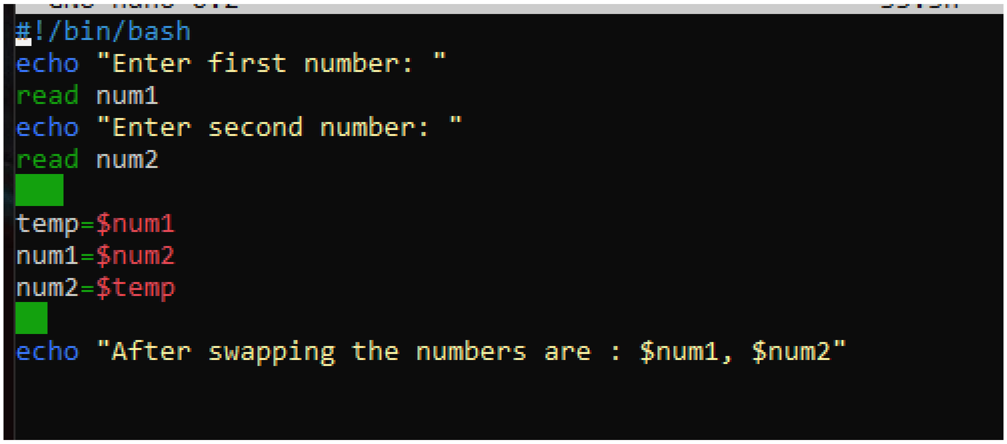


```
#!/bin/bash
echo "Enter a number"
read num
echo "obase=2; $num" | bc
```

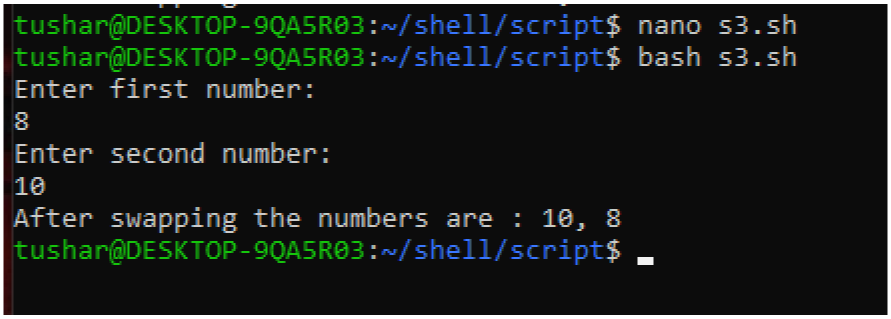


```
tushar@DESKTOP-9QA5R03:~/shell/script$ sh s.sh
Enter a number
5
101
tushar@DESKTOP-9QA5R03:~/shell/script$
```

4. Write a script to **swap** 2 numbers using intermediate variable.

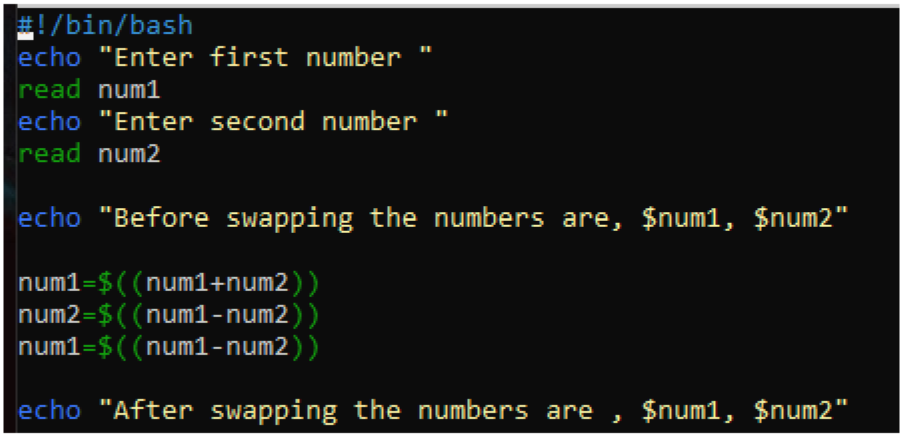


```
#!/bin/bash
echo "Enter first number: "
read num1
echo "Enter second number: "
read num2

temp=$num1
num1=$num2
num2=$temp

echo "After swapping the numbers are : $num1, $num2"
```

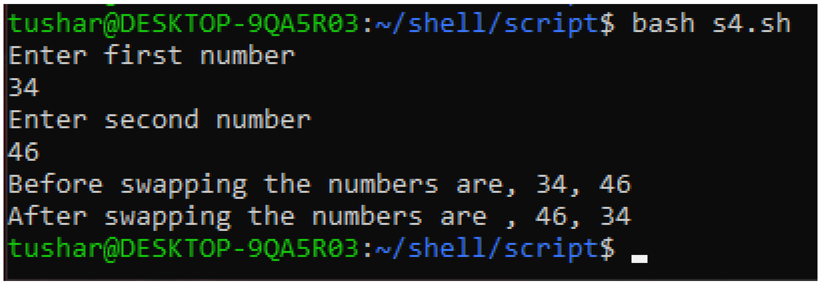


```
tushar@DESKTOP-9QA5R03:~/shell/script$ nano s3.sh
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s3.sh
Enter first number:
8
Enter second number:
10
After swapping the numbers are : 10, 8
tushar@DESKTOP-9QA5R03:~/shell/script$
```

5. Write a script to **swap** 2 numbers without using intermediate variable



```
#!/bin/bash
echo "Enter first number "
read num1
echo "Enter second number "
read num2

echo "Before swapping the numbers are, $num1, $num2"

num1=$((num1+num2))
num2=$((num1-num2))
num1=$((num1-num2))

echo "After swapping the numbers are , $num1, $num2"
```

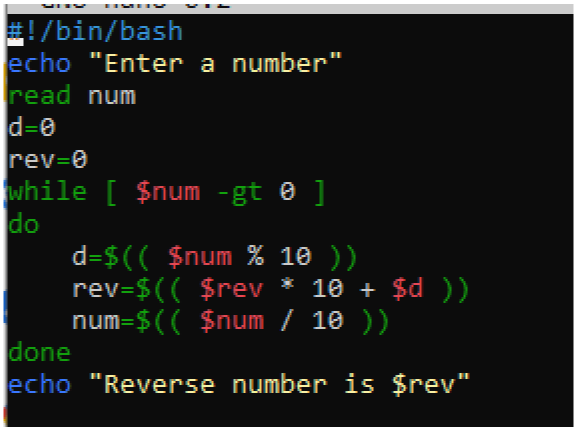


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s4.sh
Enter first number
34
Enter second number
46
Before swapping the numbers are, 34, 46
After swapping the numbers are , 46, 34
tushar@DESKTOP-9QA5R03:~/shell/script$
```

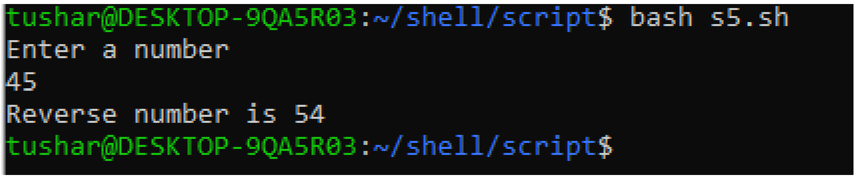
. 6. Write a script to reverse a number using while loop.

Example:

input : 12 output: 21
input : 213 output: 312
input : 125634 output:436521



```
#!/bin/bash
echo "Enter a number"
read num
d=0
rev=0
while [ $num -gt 0 ]
do
    d=$(( $num % 10 ))
    rev=$(( $rev * 10 + $d ))
    num=$(( $num / 10 ))
done
echo "Reverse number is $rev"
```

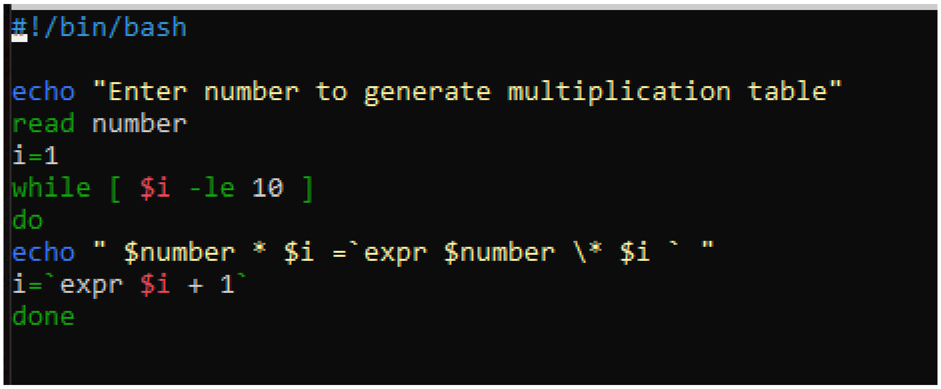


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s5.sh
Enter a number
45
Reverse number is 54
tushar@DESKTOP-9QA5R03:~/shell/script$
```

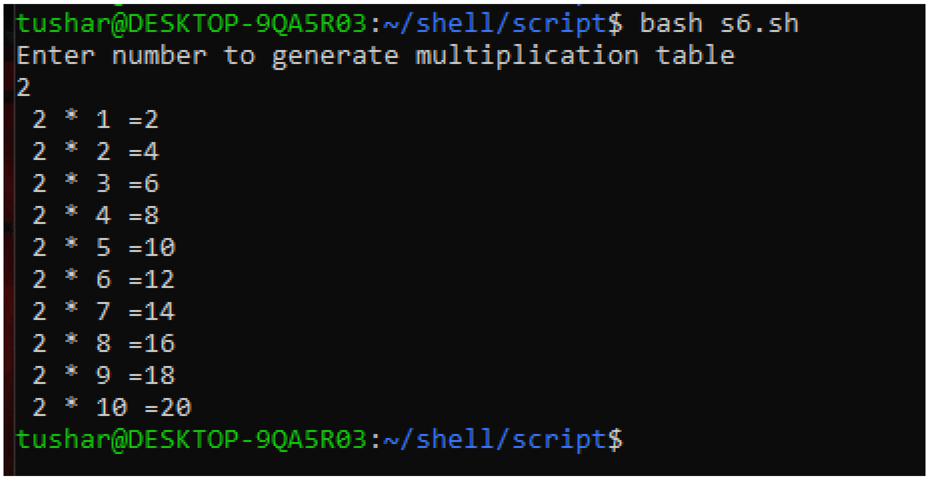
7. print multiplication table of integer using while loop.

Example:

2 x 1 = 2
2 x 2 = 4

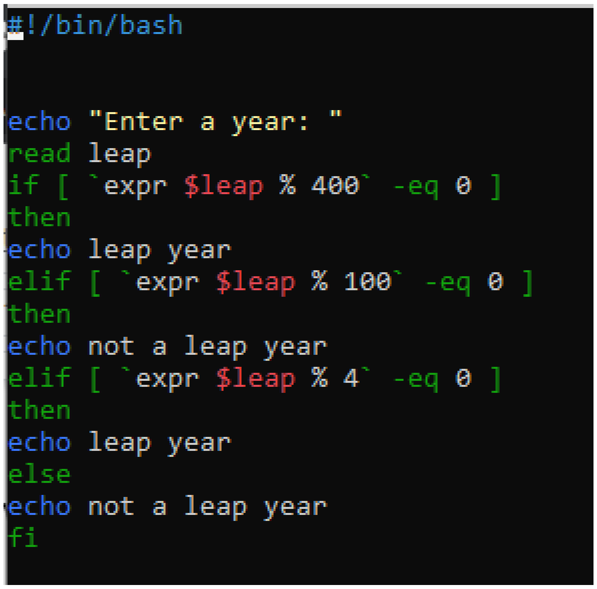


```
#!/bin/bash

echo "Enter number to generate multiplication table"
read number
i=1
while [ $i -le 10 ]
do
echo " $number * $i =`expr $number \* $i ` "
i=`expr $i + 1`
done
```

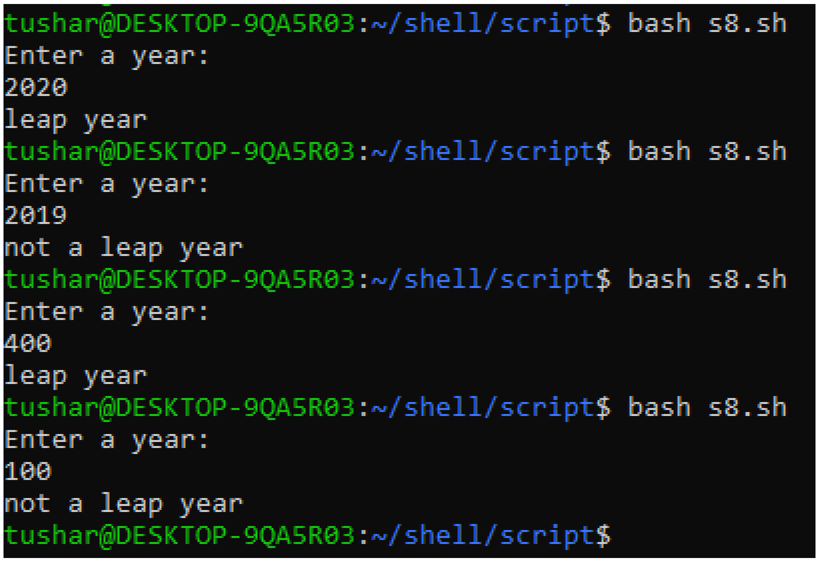


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s6.sh
Enter number to generate multiplication table
2
 2 * 1 =2
 2 * 2 =4
 2 * 3 =6
 2 * 4 =8
 2 * 5 =10
 2 * 6 =12
 2 * 7 =14
 2 * 8 =16
 2 * 9 =18
 2 * 10 =20
tushar@DESKTOP-9QA5R03:~/shell/script$
```

8. Get year as an input from user and find whether year is leap year or not.



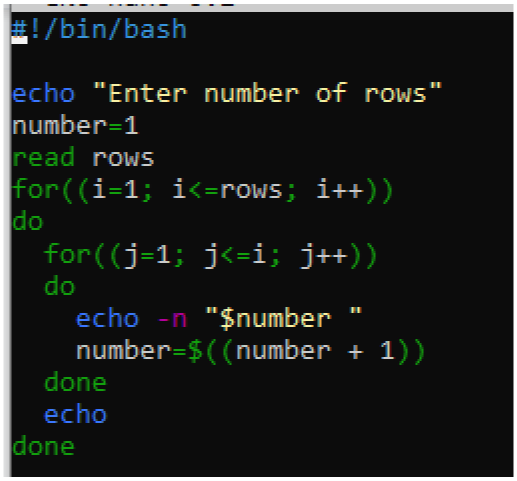
```
#!/bin/bash


echo "Enter a year: "
read leap
if [ `expr $leap % 400` -eq 0 ]
then
echo leap year
elif [ `expr $leap % 100` -eq 0 ]
then
echo not a leap year
elif [ `expr $leap % 4` -eq 0 ]
then
echo leap year
else
echo not a leap year
fi
```



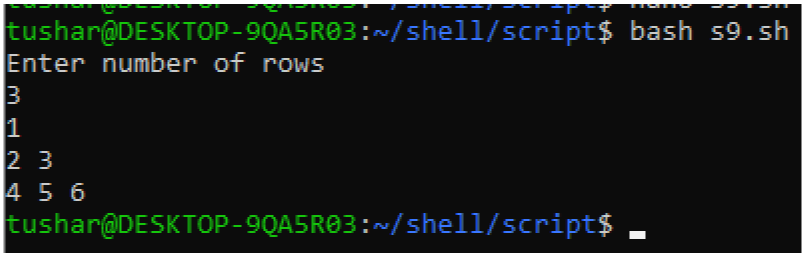
```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s8.sh
Enter a year:
2020
leap year
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s8.sh
Enter a year:
2019
not a leap year
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s8.sh
Enter a year:
400
leap year
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s8.sh
Enter a year:
100
not a leap year
tushar@DESKTOP-9QA5R03:~/shell/script$
```

9. Write a script to read the number of rows to be displayed in the pattern and print following pattern using for loop:

```
1
2 3
4 5 6
……..
```



```
#!/bin/bash

echo "Enter number of rows"
number=1
read rows
for((i=1; i<=rows; i++))
do
  for((j=1; j<=i; j++))
  do
    echo -n "$number "
    number=$((number + 1))
  done
  echo
done
```



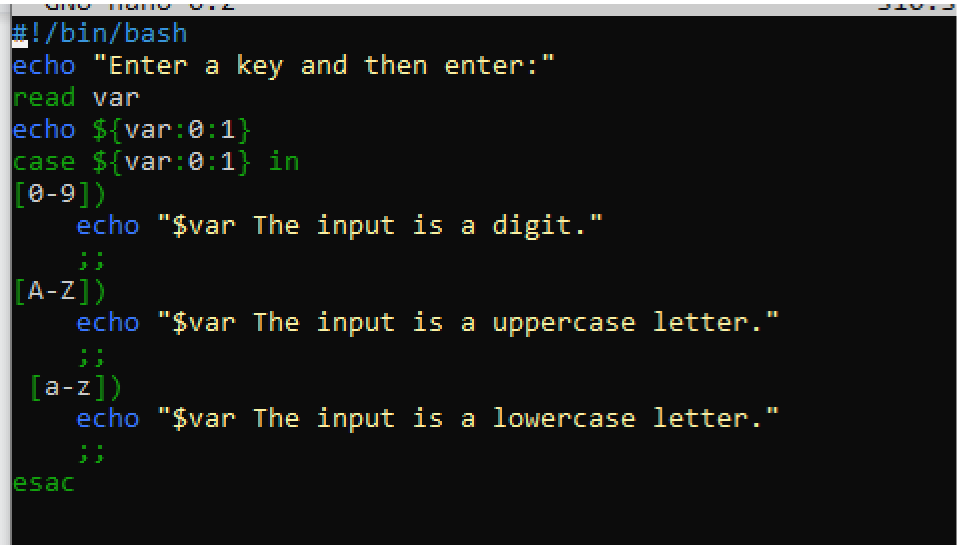
```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s9.sh
Enter number of rows
3
1
2 3
4 5 6
tushar@DESKTOP-9QA5R03:~/shell/script$
```

10. Write a script using **case** condition to do the following

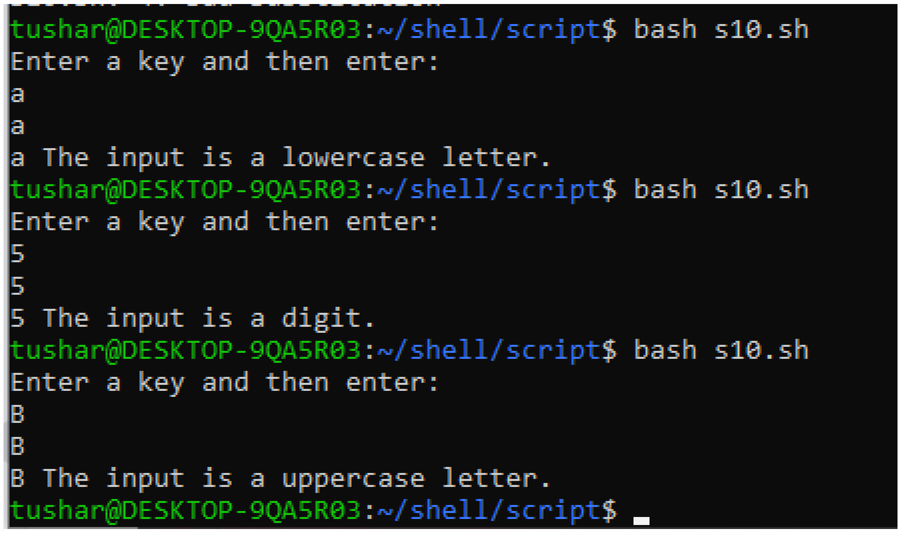
- Display “Press any key of keyboard and then press enter key”

- If the given input is number display “The input is digit.” message - If the given input is lowercase letter then display “The input is lowercase Letter.” message

- If the given input is uppercase letter then display “The input is Uppercase letter.” message

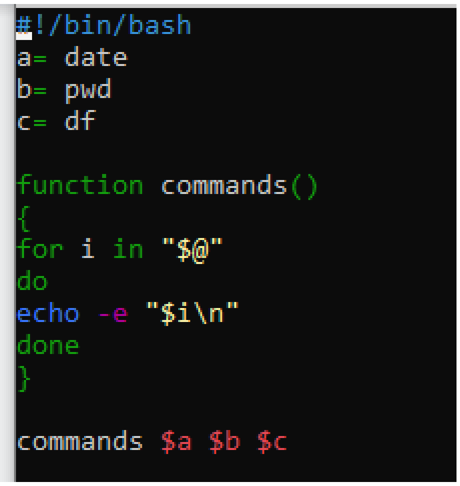


```
#!/bin/bash
echo "Enter a key and then enter:"
read var
echo ${var:0:1}
case ${var:0:1} in
[0-9])
    echo "$var The input is a digit."
    ;;
[A-Z])
    echo "$var The input is a uppercase letter."
    ;;
 [a-z])
    echo "$var The input is a lowercase letter."
    ;;
esac
```

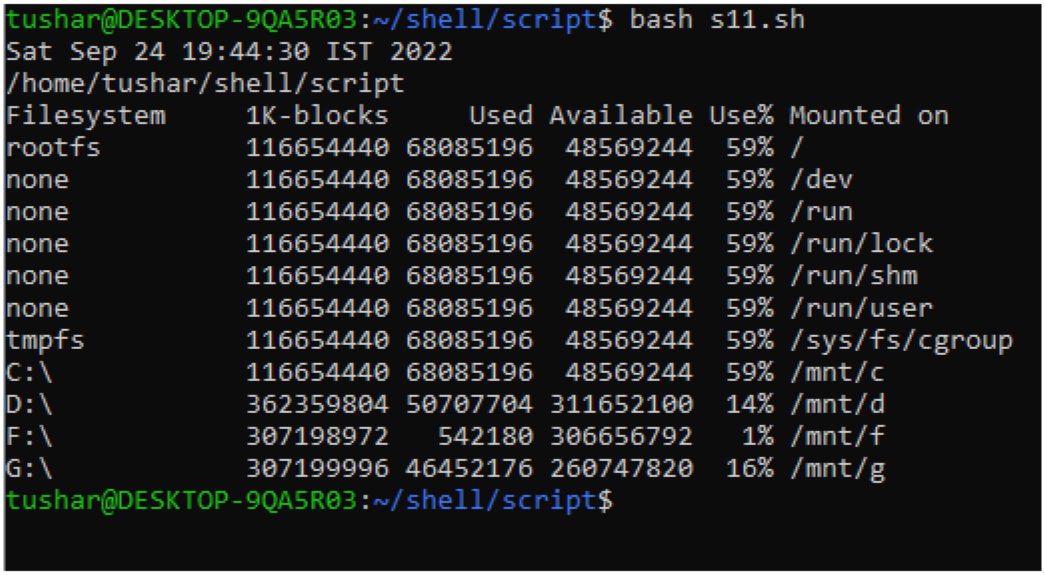


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s10.sh
Enter a key and then enter:
a
a
a The input is a lowercase letter.
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s10.sh
Enter a key and then enter:
5
5
5 The input is a digit.
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s10.sh
Enter a key and then enter:
B
B
B The input is a uppercase letter.
tushar@DESKTOP-9QA5R03:~/shell/script$ _
```

11.Write a for loop to display the outputs of **Date,pwd, df** commands. **df** command displays system disk usage details.
(**Hint:** give these commands as input to for loop)

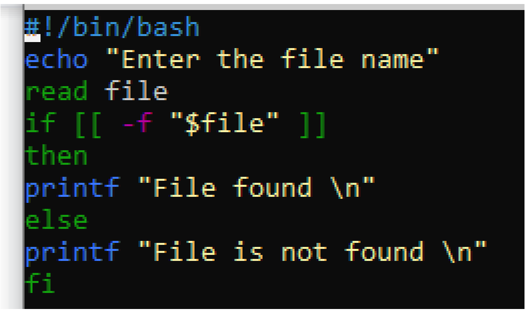


```
#!/bin/bash
a= date
b= pwd
c= df

function commands()
{
for i in "$@"
do
echo -e "$i\n"
done
}

commands $a $b $c
```

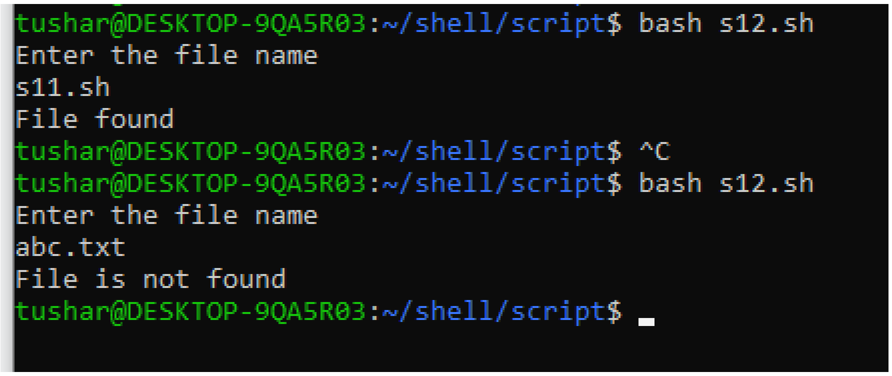


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s11.sh
Sat Sep 24 19:44:30 IST 2022
/home/tushar/shell/script
Filesystem     1K-blocks     Used Available Use% Mounted on
rootfs         116654440 68085196  48569244  59% /
none           116654440 68085196  48569244  59% /dev
none           116654440 68085196  48569244  59% /run
none           116654440 68085196  48569244  59% /run/lock
none           116654440 68085196  48569244  59% /run/shm
none           116654440 68085196  48569244  59% /run/user
tmpfs          116654440 68085196  48569244  59% /sys/fs/cgroup
C:\            116654440 68085196  48569244  59% /mnt/c
D:\            362359804 50707704 311652100  14% /mnt/d
F:\            307198972   542180 306656792   1% /mnt/f
G:\            307199996 46452176 260747820  16% /mnt/g
tushar@DESKTOP-9QA5R03:~/shell/script$
```

12.Write a script to take filename as argument and display the file exists or not
Note: If the file exists in current working directory just give filename as argument, If not give absolute path of that file as argument.



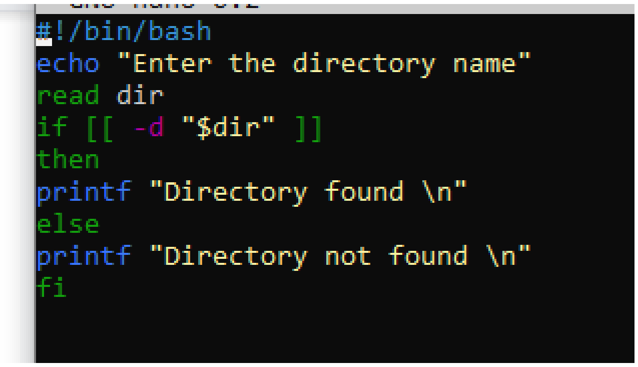
```
#!/bin/bash
echo "Enter the file name"
read file
if [[ -f "$file" ]]
then
printf "File found \n"
else
printf "File is not found \n"
fi
```



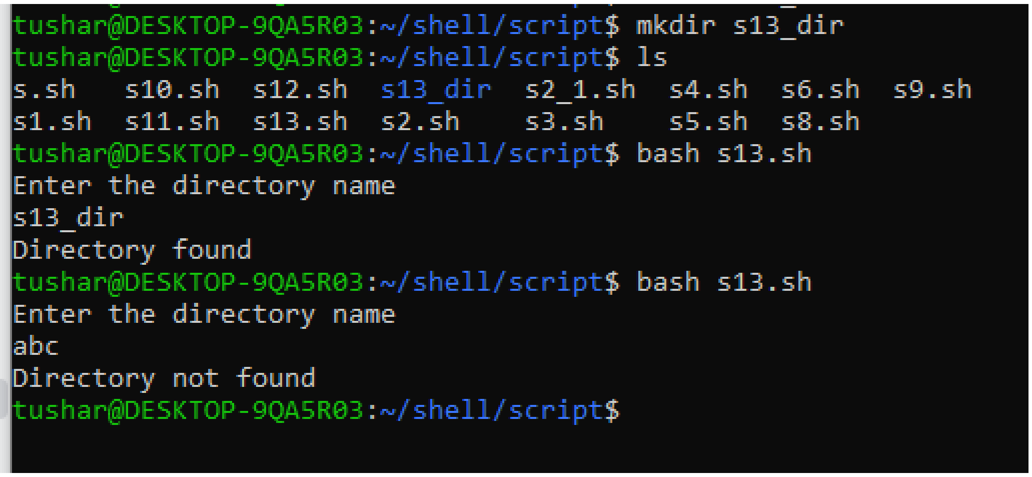
```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s12.sh
Enter the file name
s11.sh
File found
tushar@DESKTOP-9QA5R03:~/shell/script$ ^C
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s12.sh
Enter the file name
abc.txt
File is not found
tushar@DESKTOP-9QA5R03:~/shell/script$
```

13.Write a script to take directory name as argument and display the directory exists or not.

Note: If the directory exists in current working directory just give filename as argument, If not give absolute path of that file as argument.

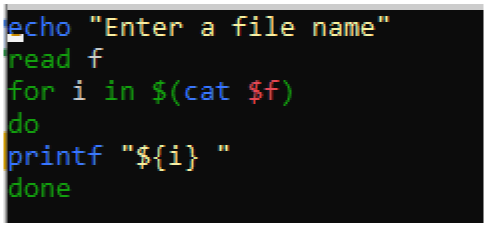


```
#!/bin/bash
echo "Enter the directory name"
read dir
if [[ -d "$dir" ]]
then
printf "Directory found \n"
else
printf "Directory not found \n"
fi
```

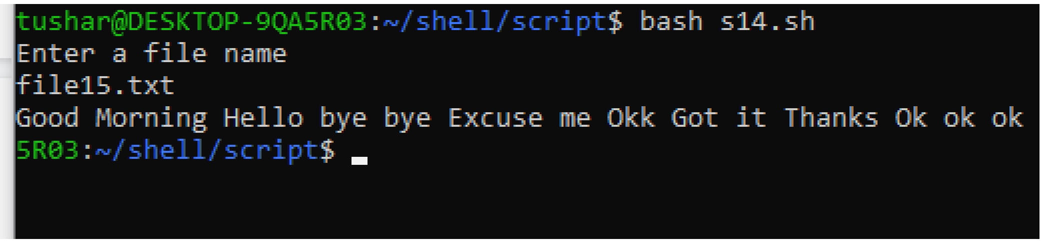


```
tushar@DESKTOP-9QA5R03:~/shell/script$ mkdir s13_dir
tushar@DESKTOP-9QA5R03:~/shell/script$ ls
s.sh   s10.sh  s12.sh  s13_dir  s2_1.sh  s4.sh  s6.sh  s9.sh
s1.sh  s11.sh  s13.sh  s2.sh    s3.sh    s5.sh  s8.sh
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s13.sh
Enter the directory name
s13_dir
Directory found
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s13.sh
Enter the directory name
abc
Directory not found
tushar@DESKTOP-9QA5R03:~/shell/script$
```

14.Read a file and display the contents of the file line by line using for loop and pass the file as command line argument to the script.

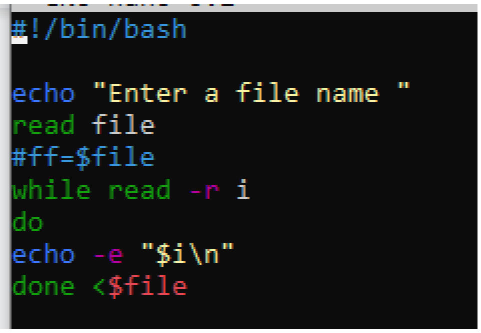


```
echo "Enter a file name"
read f
for i in $(cat $f)
do
printf "${i} "
done
```

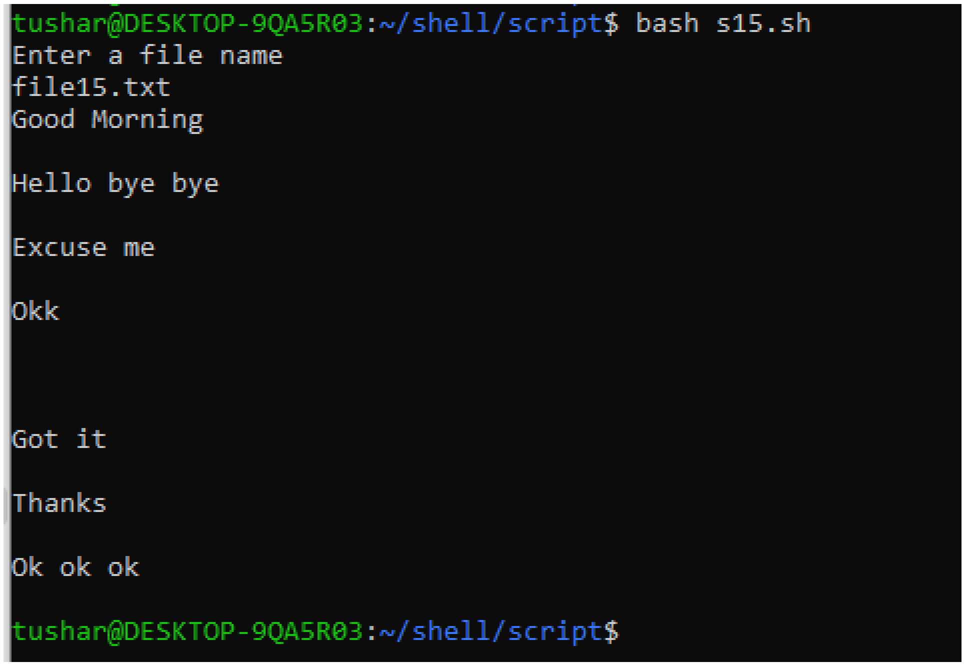


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s14.sh
Enter a file name
file15.txt
Good Morning Hello bye bye Excuse me Okk Got it Thanks Ok ok ok
5R03:~/shell/script$ _
```

15.Read a file and display the contents of the file line by line using while loop and pass the file as command line argument to the script.



```
#!/bin/bash

echo "Enter a file name "
read file
#ff=$file
while read -r i
do
echo -e "$i\n"
done <$file
```



```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s15.sh
Enter a file name
file15.txt
Good Morning

Hello bye bye

Excuse me

Okk



Got it

Thanks

Ok ok ok

tushar@DESKTOP-9QA5R03:~/shell/script$
```

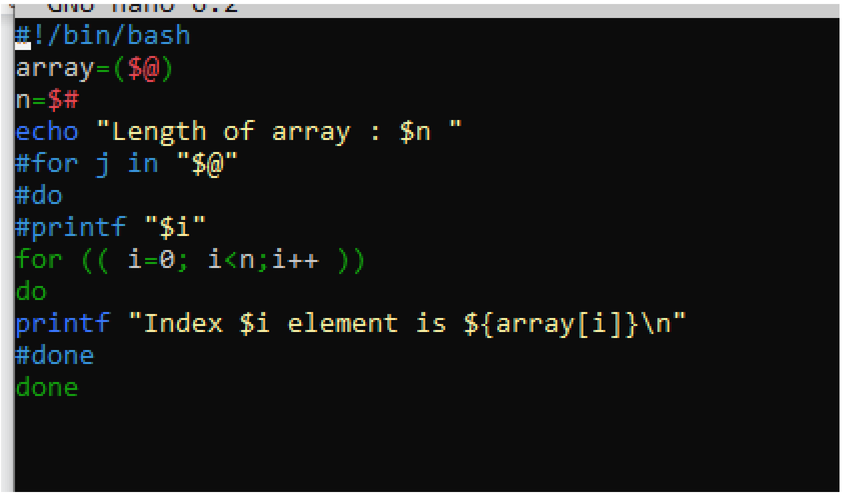
16.Write a shell script to read array elements as command line arguments assign the arguments to array and do the following

- Display the length of the array.
- Display the all elements and their index values.

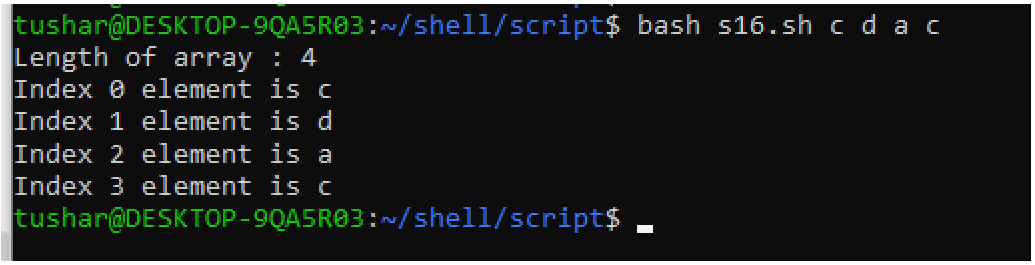
Example :

let the array elements are as follows
arr[0]=”zero”, arr[1]=”one”, arr[2]=”two”,

output should be :
length of the array : 3
index 0 element is “zero”
index 1 element is “one”
index 2 element is “two”

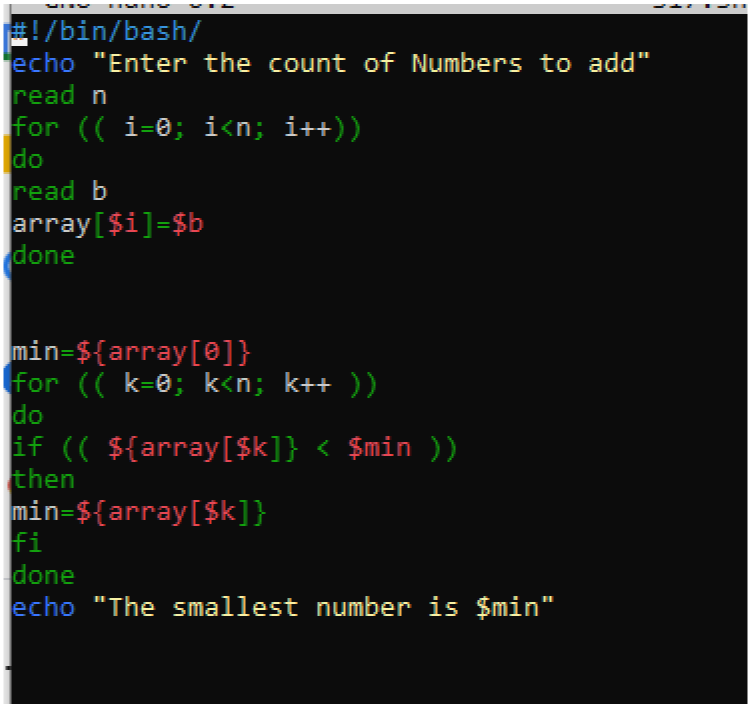


```
GNU nano 6.2
#!/bin/bash
array=($@)
n=$#
echo "Length of array : $n "
#for j in "$@"
#do
#printf "$i"
for (( i=0; i<n;i++ ))
do
printf "Index $i element is ${array[i]}\n"
#done
done
```

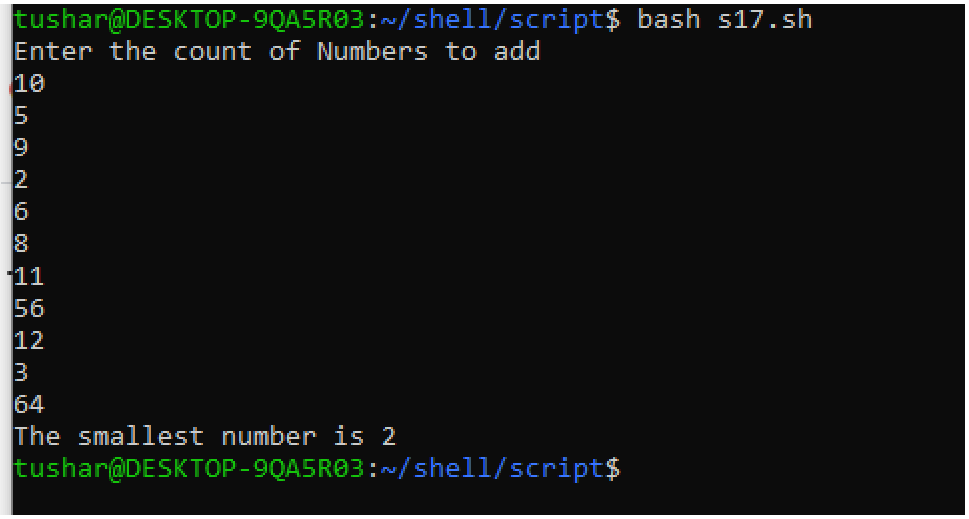


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s16.sh c d a c
Length of array : 4
Index 0 element is c
Index 1 element is d
Index 2 element is a
Index 3 element is c
tushar@DESKTOP-9QA5R03:~/shell/script$ _
```

17. Write a program to read array of 10 integers from user and find the smallest number in the array and print it.

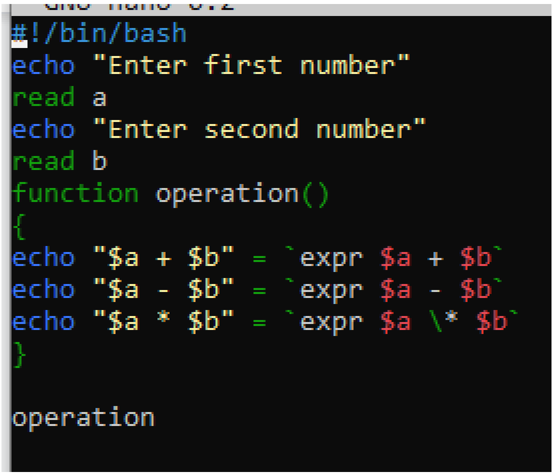


```
#!/bin/bash/
echo "Enter the count of Numbers to add"
read n
for (( i=0; i<n; i++))
do
read b
array[$i]=$b
done


min=${array[0]}
for (( k=0; k<n; k++ ))
do
if (( ${array[$k]} < $min ))
then
min=${array[$k]}
fi
done
echo "The smallest number is $min"
```

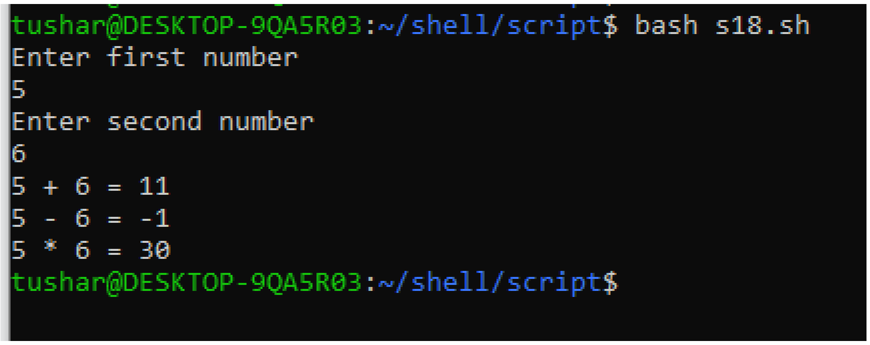


```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s17.sh
Enter the count of Numbers to add
10
5
9
2
6
8
11
56
12
3
64
The smallest number is 2
tushar@DESKTOP-9QA5R03:~/shell/script$
```

18. Write a single shell script to do the following
- read two numbers as input from terminal.
- write Add,Sub and Mul functions to perform addition,subtraction and multiplication between two integers.
    - Display the results(**Note:** To display float results using basic calculator.)



```
  GNU nano 6.2
#!/bin/bash
echo "Enter first number"
read a
echo "Enter second number"
read b
function operation()
{
echo "$a + $b" = `expr $a + $b`
echo "$a - $b" = `expr $a - $b`
echo "$a * $b" = `expr $a \* $b`
}

operation
```



```
tushar@DESKTOP-9QA5R03:~/shell/script$ bash s18.sh
Enter first number
5
Enter second number
6
5 + 6 = 11
5 - 6 = -1
5 * 6 = 30
tushar@DESKTOP-9QA5R03:~/shell/script$
```