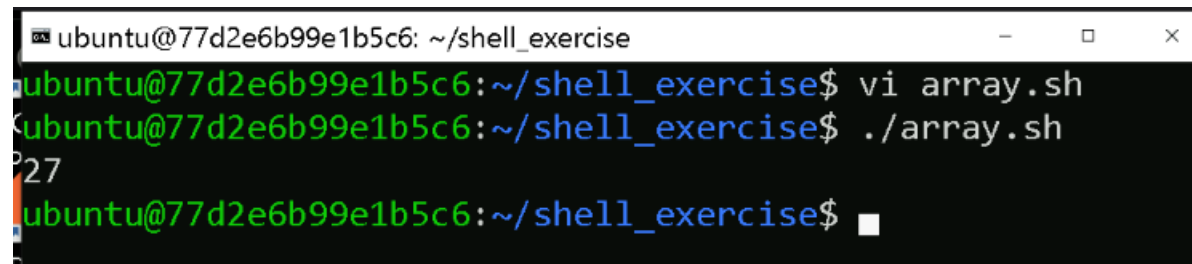Name: Barathkumar J K

Shell scripting project

1) Sum of all positive numbers in an array

```
arr=(2 4 -5 -8 9 12)
for (( i=0; i < ${#arr[*]}; i++ ));
do
        if (( arr[i] > 0 ));
        then
                sum=`expr $sum + ${arr[i]}`
        fi
done
echo "$sum"
```

```
ubuntu@77d2e6b99e1b5c6: ~/shell_exercise                    —   □   ×
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ vi array.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./array.sh
27
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ■
```

2) Reverse the number

```
#!/bin/bash
n=901
rev=0
sd=0
while [ $n -gt 0 ]
do
        sd=`expr $n % 10`
        rev=`expr $rev \* 10 + $sd`
        n=`expr $n / 10`
done
echo "Reverse Number is $rev"
```

```
buntu@77d2e6b99e1b5c6:~/shell_exercise$ vi reverse_the_numer.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ chmod u+x ./reverse_the_num
er.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ls -l
total 12
-rwxr-xr-x 1 ubuntu ubuntu 139 Jan 24 04:30 array.sh
-rwxr--r-- 1 ubuntu ubuntu 240 Jan 22 12:25 greetings.sh
-rwxr--r-- 1 ubuntu ubuntu 152 Jan 24 05:01 reverse_the_numer.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./reverse_the_numer.sh
Reverse Number is 109
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$
```

3) Palindrome number

```
echo "Enter the number:"
read n
num=$n
rev=0
while [ $n -gt 0 ]
do
        a=`expr $n % 10`
        n=`expr $n / 10`
        rev=`expr $rev \* 10 + $a`
done
echo $rev
if [ $num -eq $rev ]
then
        echo "The number is a palindrome!"
else
        echo "The number is not palindrome!"
fi
```

```
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ vi palindrome.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ chmod u+x palindrome.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./palindrome.sh
Enter the number:
901
109
The number is not palindrome!
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$
```

4) Bubble sort

```
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 arra"
        echo ${arr[@]}
~
```

```
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ vi bubble_sort.sh

ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ chmod u+x bubble_
sort.sh
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./bubble_sort.sh
Entered array:
10 8 20 100 12
sorted arra
8 10 12 20 100
```

5) Pascal Triangle

```
pastri()
{
        r=$1
        c=1
        for((i=0;i<r;i++))
            {
                    for((s=1;s<r-i;s++))
                        {
                                echo -n "  "
                        }
                    for((j=0;j<=i;j++))
                        {
                                    if [ $j -eq 0 -o $i -eq 0 ]
                                    then
                                            c=1;
                                    else
                                            c=$((c*(i-j+1)/j))
                                    fi
                                    echo -n $c "  "
                        }
                    echo
            }
        echo "enter the number of rows:"
        read r
        pastri $r
```

```
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./pascal_traingle
.sh
enter the number of rows:
4
       1
     1   1
   1   2   1
 1   3   3   1
```

6) Shell program to add two integer values and check if any input is given or not

```
#/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 $inp2 ]]
then
        echo "Input 2 cannot be empty, please enter an integer."
fi
bc_val=`echo "$inp1+$inp2" | bc`
echo "BC value: $bc_val"

expr_val=`expr $inp1 + $inp2`
echo "Expr value : $expr_val"
~
```

```
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./arithmetic.sh
Input1: 40
Input2: 40
BC value: 80
Expr value : 80
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$
```

7) Hello greetings

```
#!/bin/bash

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

```
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ chmod 700 name_greetings.s
h
ubuntu@77d2e6b99e1b5c6:~/shell_exercise$ ./name_greetings.sh
barath
Hello, barath! Welcome to the world of shell scripting!
```