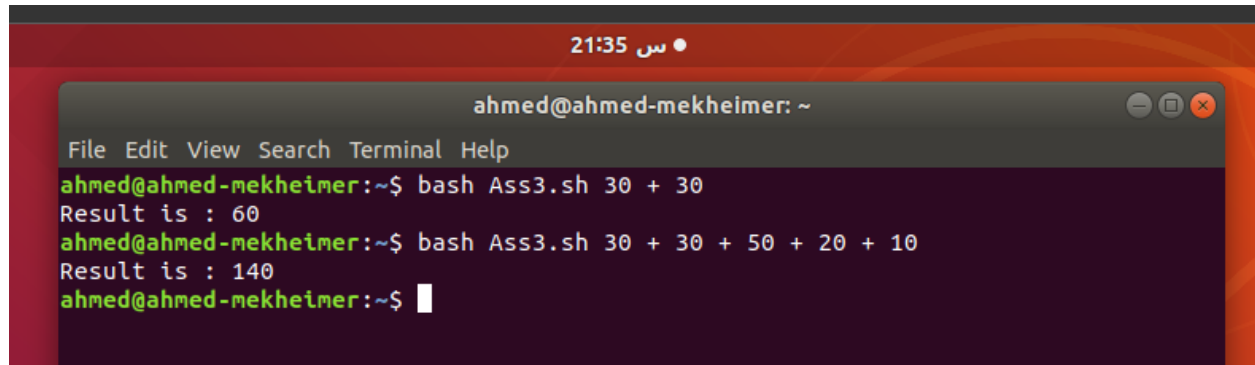


Ass 3

Test Cases:

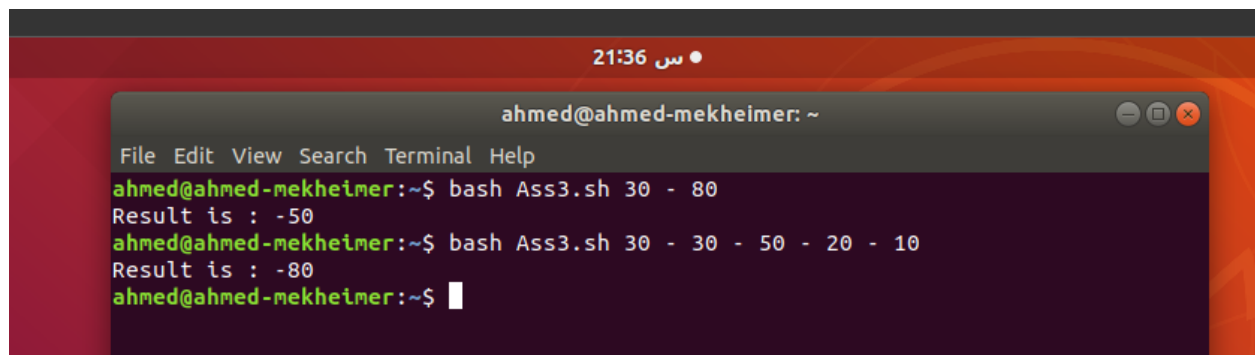
Addition



A terminal window titled 'ahmed@ahmed-mekheimer: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows two test cases for addition. The first command is 'bash Ass3.sh 30 + 30', which outputs 'Result is : 60'. The second command is 'bash Ass3.sh 30 + 30 + 50 + 20 + 10', which outputs 'Result is : 140'. The prompt 'ahmed@ahmed-mekheimer:~\$' is visible at the end of each line.

```
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 + 30
Result is : 60
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 + 30 + 50 + 20 + 10
Result is : 140
ahmed@ahmed-mekheimer:~$
```

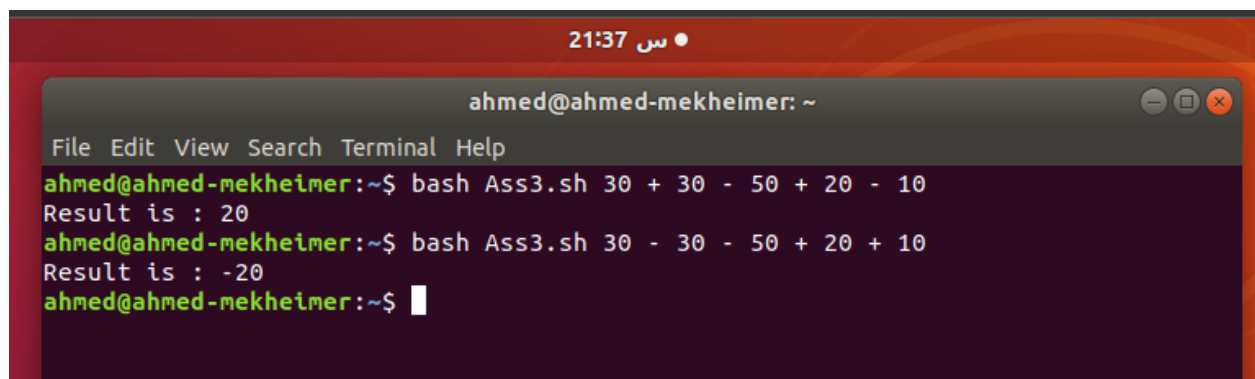
Subtraction



A terminal window titled 'ahmed@ahmed-mekheimer: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows two test cases for subtraction. The first command is 'bash Ass3.sh 30 - 80', which outputs 'Result is : -50'. The second command is 'bash Ass3.sh 30 - 30 - 50 - 20 - 10', which outputs 'Result is : -80'. The prompt 'ahmed@ahmed-mekheimer:~\$' is visible at the end of each line.

```
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 80
Result is : -50
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 30 - 50 - 20 - 10
Result is : -80
ahmed@ahmed-mekheimer:~$
```

Multiple addition and subtraction



A terminal window titled 'ahmed@ahmed-mekheimer: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows two test cases for multiple addition and subtraction. The first command is 'bash Ass3.sh 30 + 30 - 50 + 20 - 10', which outputs 'Result is : 20'. The second command is 'bash Ass3.sh 30 - 30 - 50 + 20 + 10', which outputs 'Result is : -20'. The prompt 'ahmed@ahmed-mekheimer:~\$' is visible at the end of each line.

```
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 + 30 - 50 + 20 - 10
Result is : 20
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 30 - 50 + 20 + 10
Result is : -20
ahmed@ahmed-mekheimer:~$
```

```
22:07 س ●
ahmed@ahmed-mekheimer: ~
File Edit View Search Terminal Help
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 30 - 50 + 20 + 20 - 20 + 10 - 40
Result is : -60
ahmed@ahmed-mekheimer:~$
```

Invalid Number of Arguments:

```
21:41 س ●
ahmed@ahmed-mekheimer: ~
File Edit View Search Terminal Help
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 30 - 50 + 20 +
Invalid Number of arguments, must be 3 or greater and odd
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 - 30 -
Invalid Number of arguments, must be 3 or greater and odd
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 30 -
Invalid Number of arguments, must be 3 or greater and odd
ahmed@ahmed-mekheimer:~$
```

Invalid input in arguments

```
21:58 س ●
ahmed@ahmed-mekheimer: ~
File Edit View Search Terminal Help
ahmed@ahmed-mekheimer:~$ bash Ass3.sh = - 20
Invalid input in 1 st Number: Numbers(+ve/-ve/Float) only.
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 - !
Invalid input in 2 nd Number: Numbers only.
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 x 20
Invalid 1 st Operator
ahmed@ahmed-mekheimer:~$
```

```
22:00 س ●
ahmed@ahmed-mekheimer: ~
File Edit View Search Terminal Help
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 + 20 - =
Invalid input of 3 rd/th Number: Numbers(+ve/-ve/Float) only.
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 + 20 x 20
Invalid 2 st/nd/th Operator
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 + 20 + 20 / 30
Invalid 3 st/nd/th Operator
ahmed@ahmed-mekheimer:~$ bash Ass3.sh 20 + 20 + 20 - x
Invalid input of 4 rd/th Number: Numbers(+ve/-ve/Float) only.
ahmed@ahmed-mekheimer:~$
```

Bash file:

Drive link for Ass.3 txt file

<https://drive.google.com/drive/folders/1zf2OjlzdJCJFI7bAUkXuDuMv6XiZDZ08?usp=sharing>

```
ies Text Editor 22:04 س
Ass3.sh
~/
#!/bin/bash
numofarg=$#;
numofarg1=$(( $numofarg%2 ));
x=0
sum=0
if ! [ $# -ge 3 ]
then
    echo "Invalid Number of arguments, must be 3 or greater and odd"
    exit 1
fi
if ! [ $numofarg1 != $x ]
then
    echo "Invalid Number of arguments, must be 3 or greater and odd"
    exit 1
else
    i=1
    input1=1
    input2=2
    inop=1
    while [[ $i -le $numofarg ]]
    do
        #echo "Entered while"
        if [ $i -eq 1 ]
        then
            num1=$1
            shift 1
            #input1=$((input1 + 2));
            operator=$1
            shift 1
            num2=$1
            shift 1
            #echo "Before f()"
```

```

#echo "Before f()"
#function that tests and calculates
if ! [[ $num1 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
then
    echo "Invalid input in $input1 st Number: Numbers(+ve/-ve/Float) only."
    exit 1

elif ! [[ $num2 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
then
    echo "Invalid input in $input2 nd Number: Numbers only."
    exit 1
fi

case $operator in
+) sum=$(( num1 + num2 ))
    #echo "+"
;;
-) sum=$(( num1 - num2 ))
    #echo "-"
;;
*) echo "Invalid $inop st Operator"
    exit 1
;;
esac
#echo "i++"
i=$((i + 3));

else
    l=$((l + 3));

else
    #echo "Entered 2nd phase"
    operator=$1
    shift 1
    num2=$1
    inop=$((inop + 1));
    input2=$((inop + 1));
    if ! [[ $num2 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
    then
        echo "Invalid input of $input2 rd/th Number: Numbers(+ve/-ve/Float) only."
        exit 1
    fi

    case $operator in
+) sum=$(( sum + num2 ))
        #echo "+"
;;
-) sum=$(( sum - num2 ))
        #echo "-"
;;
*) echo "Invalid $inop st/nd/th Operator"
        exit 1
;;
esac
    shift 1
    i=$((i + 2))
    #echo "i++"
fi

#echo "l++"
l=$((l + 2))
fi

done
#echo "END WHILE"
fi
#echo "END IF"
echo "Result is : $sum"

```

```
#!/bin/bash
numofarg=$#;
numofarg1=$(( $numofarg%2 ));
x=0
sum=0
if ! [ $# -ge 3 ]
then
    echo "Invalid Number of arguments, must be 3 or greater and odd"
    exit 1
fi
if ! [ $numofarg1 != $x ]
then
    echo "Invalid Number of arguments, must be 3 or greater and odd"
    exit 1
else
    i=1
    input1=1
    input2=2
    inop=1
    while [[ $i -le $numofarg ]]
    do
        #echo "Entered while"
        if [ $i -eq 1 ]
        then
            num1=$1
            shift 1
            #input1=$((input1 + 2));
```

only."

```
operator=$1
shift 1
num2=$1
shift 1
#echo "Before f()"
#function that tests and calculates
if ! [[ $num1 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
then
    echo "Invalid input in $input1 st Number: Numbers(+ve/-ve/Float)

    exit 1

elif ! [[ $num2 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
then
    echo "Invalid input in $input2 nd Number: Numbers only."
    exit 1
fi

case $operator in
    +) sum=$(( num1 + num2 ))
        #echo "+"
        ;;
    -) sum=$(( num1 - num2 ))
        #echo "-"
        ;;
    *) echo "Invalid $inop st Operator"
        exit 1
```

```

        ;;
    esac
    #echo "i+++"
    i=$((i + 3));
else
    #echo "Entered 2nd phase"
    operator=$1
    shift 1
    num2=$1
    inop=$((inop + 1));
    input2=$((inop + 1));
    if ! [[ $num2 =~ ^[+-]?[0-9]+([.][0-9]+)?$ ]] ;
    then
        echo "Invalid input of $input2 rd/th Number: Numbers(+ve/-
ve/Float) only."
        exit 1
    fi

    case $operator in
        +) sum=$(( sum + num2 ))
            #echo "+"
            ;;
        -) sum=$(( sum - num2 ))
            #echo "-"
            ;;
        *) echo "Invalid $inop st/nd/th Operator"

```

```
        exit 1
    ;;
esac
shift 1
i=$((i + 2))
#echo "i++"
fi
done
#echo "END WHILE"
fi
#echo "END IF"
echo "Result is : $sum"
```