

تمرین ها:

-1

ابتدا یک فایل با پسوند sh. میسازیم. سپس کد زیر را با توجه به خواسته های دستورکار می نویسیم. و با دستور bash script.sh فایل را اجرا می کنیم.

```

1 #!/bin/bash
2
3 flag=0
4 function checkInput(){
5 if [ -z $1 ]; then
6     echo "First input is empty"
7     flag=1
8 elif ! [[ $1 =~ ^[+-]?[0-9]+$ ]]; then
9     echo "First input is not valid"
10    flag=1
11 fi
12 if [ -z $2 ]; then
13     echo "Second input is empty"
14     flag=1
15 elif ! [[ $2 =~ ^[+-]?[0-9]+$ ]]; then
16     echo "Second input is not valid"
17     flag=1
18 fi
19 }
20
21 checkInput $1 $2
22 if [ $flag -eq 0 ]; then
23     echo "=>Sum:"
24     expr $1 + $2
25
26     echo "=>Comparison:"
27     if [ $1 -gt $2 ]
28     then
29         echo $1
30     elif [ $1 -lt $2 ]
31     then
32         echo $2
33     elif [ $1 -eq $2 ]
34     then
35         echo "equal"
36     fi
37 fi
38

```

تا مفسر دستور bash را اجرا کند

این قسمت از کد اگر کاربر ورودی های اشتباه وارد کند راهنمای مناسب به آن را echo چاپ می کند. (با استفاده از دستور) مثلا اگر عملوند دوم را وارد نکند می گوید که عملوند دوم خالی می باشد. متغیر flag به یک تغییر می کند. \$1 ورودی اول و \$2 ورودی دوم است.

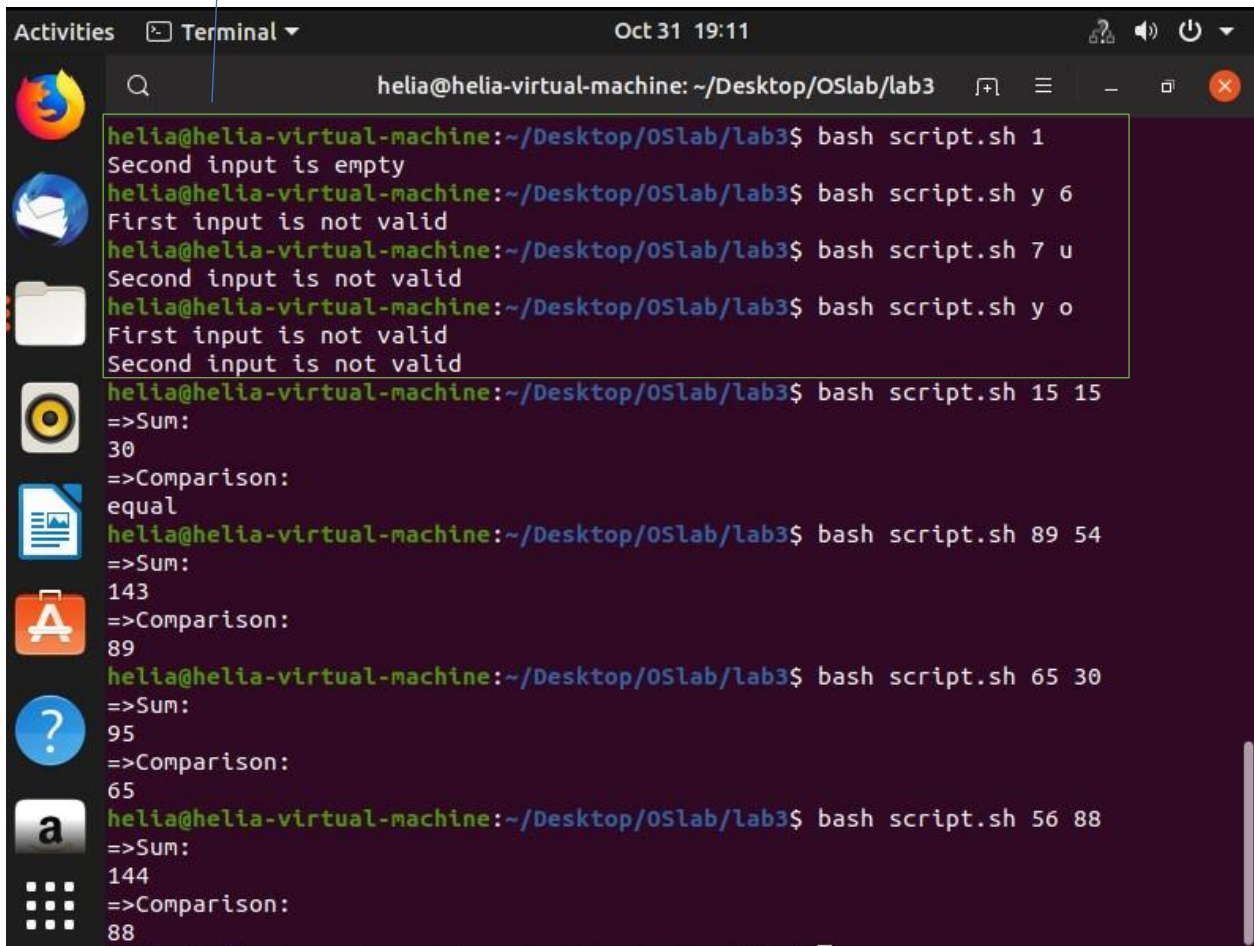
پایان if

جمع دو آرگومان ورودی را به ما می دهد

عدد بزرگتر را نمایش می دهد یا برابری آن ها را به ما نشان می دهد.

ورودی های نامطلوب

تست کد



The image shows a terminal window titled "Terminal" with a search bar and a dropdown menu. The window displays the output of a script named "script.sh" being executed in a virtual machine environment. The prompt is "helia@helia-virtual-machine: ~/Desktop/OSlab/lab3". The script is being tested with various inputs, and the output shows the sum and comparison of the inputs. A blue arrow points from the text "ورودی های نامطلوب" to the terminal window.

```
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 1
Second input is empty
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh y 6
First input is not valid
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 7 u
Second input is not valid
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh y o
First input is not valid
Second input is not valid
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 15 15
=>Sum:
30
=>Comparison:
equal
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 89 54
=>Sum:
143
=>Comparison:
89
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 65 30
=>Sum:
95
=>Comparison:
65
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script.sh 56 88
=>Sum:
144
=>Comparison:
88
```

فایل `bash` را اجرا می کنیم و کد آن با توجه به خواسته ی دستورکار به شکل زیر است. در اصل بخش چک کردن ورودی های مطلوب همچون قسمت قبل است، فقط برای عمل کردن همچون یک ماشین حساب که شامل عملیات های جمع و تفریق و ضرب و تقسیم است کدش را می نویسیم.

```

1#!/bin/bash
2
3flag=0
4function checkInput(){
5if [ -z $1 ]; then
6    echo "First number is empty"
7    flag=1
8elif ! [[ $1 =~ ^[+-]?[0-9]+$ ]]; then
9    echo "First number is not valid"
10    flag=1
11fi
12if [ -z $3 ]; then
13    echo "Second number is empty"
14    flag=1
15elif ! [[ $3 =~ ^[+-]?[0-9]+$ ]]; then
16    echo "Second number is not valid"
17    flag=1
18fi
19}
20
21checkInput $1 $2 $3
22if [ $flag -eq 0 ]; then
23    case $2 in
24        "+")
25        expr $1 + $3
26        ;;
27        "-")
28        expr $1 - $3
29        ;;
30        "x")
31        expr $1 \* $3
32        ;;
33        "/" )
34        expr $1 / $3
35        ;;
36        *)
37        echo "invalid input"
38        ;;
39    esac
40fi

```

ورودی های نامطلوب را چک می کند.

تابع صدا زده می شود و ورودی ها را می گیرد.

شروع case و چک کردن حالت های مختلف ورودی دوم که ورودی دوم نوع عملیات را نشان می دهد و در آخر هم اگر ورودی دوم هیچ یک از عملیات های قابل قبول نباشد پیام مناسب چاپ می شود.

پایان case

تست ورودی

```
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script2.sh 64 + 12
76
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script2.sh 35 - 9
26
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script2.sh 56 / 2
28
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script2.sh 41 x 4
164
```

مانند بخش های قبل فایل را ساخته و کد زیر به این صورت کار می کند که با گرفتن یک عدد به صورت ورودی حاصل جمع ارقام و برعکس عدد را به ما برمی گرداند. از `while` استفاده می کنیم (`always true` است) برای گرفتن ورودی هم از `read -p` استفاده کرده ایم. الگوریتم هم به این صورت است که ابتدا باقی مانده ی ورودی را به 10 حساب کرده سپس با ارقام قبلی آن را جمع می کنیم و در 10 ضرب می کنیم تا عدد معکوس شده به دست آید.

```

1#!/bin/bash
2
3while true; do
4    read -p 'Enter a number: ' number
5
6    digit=0
7    newNum=0
8    sumDigits=0
9
10    while [ $number -gt 0 ]
11    do
12        digit=$(( $number % 10 ))
13        sumDigits=$(( $digit + $sumDigits))
14        newNum=`expr $newNum \* 10 + $digit`
15        number=$(( $number / 10 ))
16    done
17
18    echo "Reverse number is: $newNum"
19    echo "Sum of digits is: $sumDigits"
20done
21

```

گرفتن ورودی

رقم فعلی

عدد جدید

جمع ارقام

حلقه تا زمانی که عدد ما پس از تقسیم بر 10 صفر نشود ادامه می یابد.

چاپ خروجی ها

تست ورودی

```
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script3.sh
Enter a number: 452
Reverse number is 254
Sum of digits is 11
Enter a number: 8763
Reverse number is 3678
Sum of digits is 24
```

همچون قسمت های پیشین فایل را ساخته و با دستور `read -p` شماره خط ها و نام فایل مورد نظر را به عنوان ورودی از ترمینال دریافت می کنیم. و در فایل پیمایش صورت گرفته تا شماره خط های مورد نظر پیدا شود.

```

1 #!/bin/bash
2
3 read -p 'Enter X: ' x
4 read -p 'Enter Y: ' y
5 read -p 'Enter file name: ' fileName
6
7 fileAddress="./"
8 fileAddress+="$fileName"
9 counter=1
10
11 while IFS= read -r line; do
12     if ([ $counter -gt $x ] || [ $counter -eq $x ]) && [ $counter
13         echo "$line"
14     fi
15     ((++counter))
16 done < "$fileAddress"
17

```

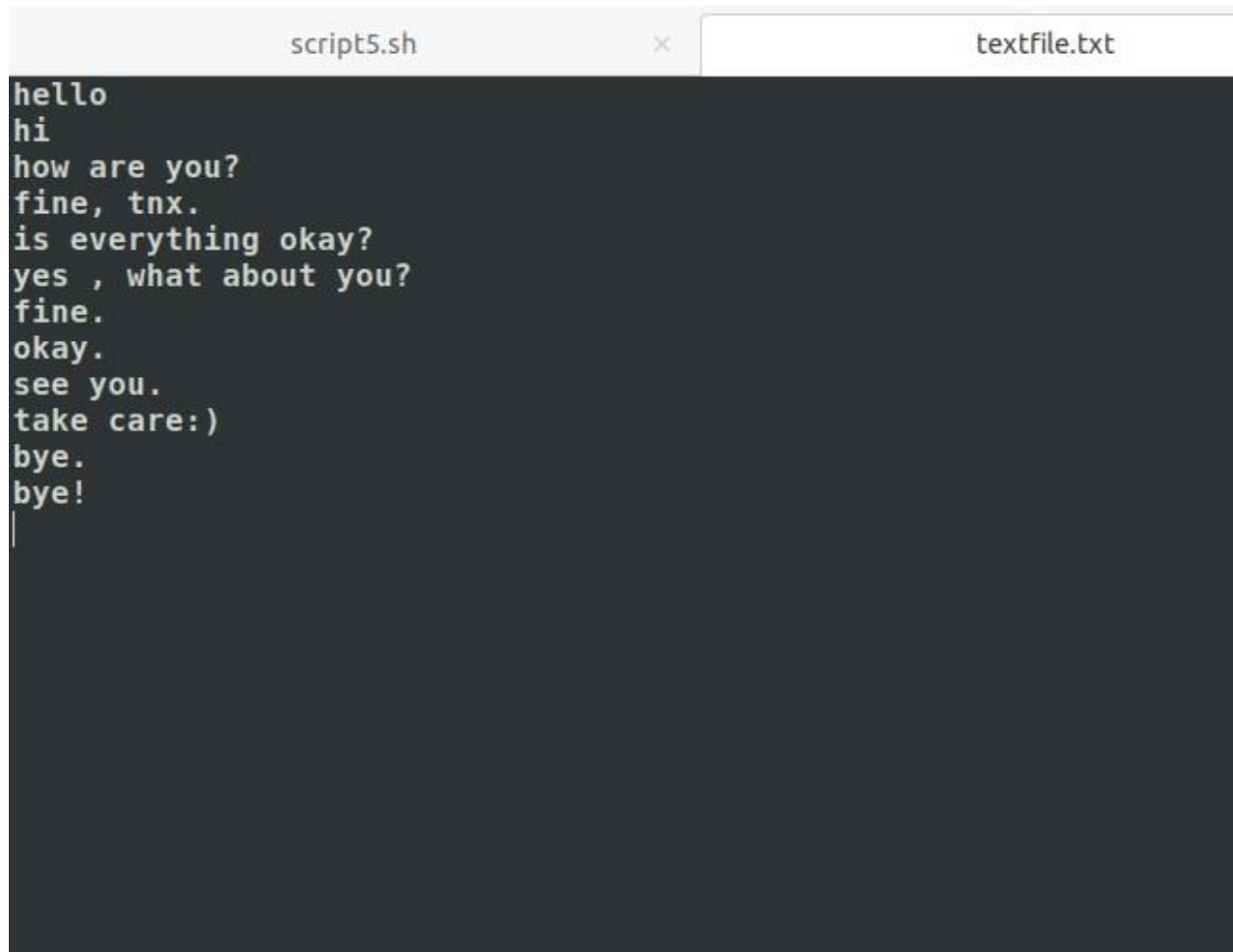
گرفتن ورودی

رشته ی آدرس را ساخته تا در همان دایرکتوریمان آن را بیابیم

اگر در خط خواسته شده بود که آن را چاپ می کند

تست ورودی

```
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ touch textfile.txt
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script4.sh
Enter X: 3
Enter Y: 5
Enter file name: textfile.txt
how are you?
fine, tnx.
is everything okay?
```



```
hello
hi
how are you?
fine, tnx.
is everything okay?
yes , what about you?
fine.
okay.
see you.
take care:)
bye.
bye!
```


همچون قسمت های پیشین فایل را ساخته و با دستور `read -p` یک ورودی گرفته تا از سه تابع مورد نظر یکی

را صدا بزنیم.

```

1#!/bin/bash
2
3function first(){
4    for i in 1 2 3 4 5
5    do
6        for j in $(seq 1 $i)
7        do
8            echo -n "$i "
9        done
10       echo
11    done
12}
13
14function second(){
15    for i in 1 2 3 4 5 6
16    do
17        for ((j=5;j>=i;j--))
18        do
19            echo -n " "
20        done
21        for j in $(seq 1 $i)
22        do
23            echo -n "* "
24        done
25        echo
26    done
27    for i in 1 2 3 4 5 6
28    do
29        for ((j=1;j<$i;j++))
30        do
31            echo -n " "
32        done
33        for ((j=$i;j<=6;j++))
34        do
35            echo -n "* "
36        done
37        echo
38    done
39}
40}
41
42function third(){
43    for ((i=1;i<=5;i++))
44    do
45
46        for ((j=1;j<$i;j++))
47        do
48            echo -n "| "
49        done
50        echo "|_"
51    done
52}
53
54case $1 in
55    "1")
56        first
57        ;;
58    "2")
59        second
60        ;;
61    "3")
62        third
63        ;;
64esac
65

```

از 1 تا i می رود

نیمه ی بالا

نیمه ی پایین

ورودی اول یا 1 یا 2 یا 3
باشد

تست ورودی

```
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script5.sh 1
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script5.sh 2
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *
* * * * * *
* * * * *
 * * * *
  * * *
   * *
    *
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script5.sh 3
|
|_
| |
| |
|_|
|_|
|_|
|_|
|_|
|_|
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$
```

امتیازی

به طور کلی ماشین حساب حقیقی با ماشین حساب صحیح تفاوتی ندارد و فقط برای بخش اعشاری دچار مشکل نمی شود.

```
1#!/bin/bash
2
3read -p "operand1= " op1
4read -p "operator=>(/ * - + )" operand
5read -p "operand2= " op2
6
7echo -n "res: "
8
9case $operand in
10    /)
11        expr "scale=4;$op1 / $op2" |bc;;
12    \*)
13
14        expr "$op1 * $op2" |bc;;
15    +)
16        expr "$op1 + $op2" |bc;;
17    -)
18        expr "$op1 - $op2" |bc;;
19esac
```

از scale=4 برای تقسیم استفاده کرده تا محاسبات را تا چهار رقم اعشار انجام دهد

برای اینکه دچار مشکل در اعداد اعشاری نشویم از bc استفاده شده است.

کار نمایش

تست ورودی

```
helia@helia-virtual-machine: ~/Desktop/OSlab/lab3
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script6.sh
operand1= 253.3
operator=>(/ * - + ) -
operand2= 0.3
res: 253.0
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script6.sh
operand1= 78.34
operator=>(/ * - + )+
operand2= 14.56
res: 92.90
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script6.sh
operand1= 56.1
operator=>(/ * - + )*
operand2= 2.2
res: 123.42
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$ bash script6.sh
operand1= 67.2
operator=>(/ * - + )/
operand2= 4.2
res: 16.0000
helia@helia-virtual-machine:~/Desktop/OSlab/lab3$
```