

به نام خدا

محمد جواد زندیه

9831032

گزارش آزمایش 3 سیستم عامل

first.sh

```
home > javad > Desktop > $ first.sh > ...
1  #!/bin/bash
2
3  # regular expretion to chech that the input is a number or not
4  re='^[0-9]+$'
5
6  # get first number
7  read -r -p 'Enter first number: ' a
8  while ! [[ $a =~ $re ]]
9  do
10 |   read -r -p 'please enter a number nothing else: ' a
11 | done
12
13 # ger second number
14 read -r -p 'Enter second number: ' b
15 while ! [[ $b =~ $re ]]
16 do
17 |   read -r -p 'please enter a number nothing else: ' b
18 | done
19
20 # compute sum
21 (( sum=a+b ))
22 echo "sum: $sum"
23
24 # find which one is greater
25 if [[ $a -gt $b ]]
26 then
27 |   echo "$a is greater than $b"
28 | else
29 |   echo "$b is greater than $a"
30 | fi
```

در این shell script فایل ابتدا دو عدد از ورودی گرفته ایم و تا زمانی که ورودی کاربر به صورت عدد نبوده است از او خواسته ایم که به درستی ورودی را به صورت یک عدد بدهد.

سپس جمع دو عدد ورودی را echo کرده ایم و پس از آن با استفاده از if, else عدد و به کاربردن -lt به معنای کوچکتر و -gt به معنای بزرگ تر، عدد بزرگتر را تشخیص داده ایم و به کاربر نشان داده ایم.

یک نمونه از خروجی:

```
javad@javad-HP-350-G1: ~/Desktop
javad@javad-HP-350-G1:~/Desktop$ ls
calculator.sh  first.sh  project3  shekl.sh
file_cat.sh   OS_LAB_InstructionManual_1398_v02.pdf  reverse.sh  test_file
javad@javad-HP-350-G1:~/Desktop$ ./first.sh
Enter first number: 54
Enter second number: fa
please enter a number nothing else: 3
sum: 57
54 is greater than 3
javad@javad-HP-350-G1:~/Desktop$
```

calculator.sh

```
home > javad > Desktop > $ calculator.sh > ...
1  #!/bin/bash
2
3  re='^[0-9]+$'
4  while true; do
5      # enter first number
6      read -r -p "Enter first number: " a
7      while ! [[ $a =~ $re ]]
8      do
9          read -r -p 'please enter a number nothing else: ' a
10         done
11
12         # enter second number
13         read -r -p 'Enter second number: ' b
14         while ! [[ $b =~ $re ]]
15         do
16             read -r -p 'please enter a number nothing else: ' b
17         done
18
19         PS3="choos operation please: "
20         select ans in add subtract multiply divide; do
21             case $ans in
22                 add)
23                     op='+' ;|
24                     break ;;
25                 subtract)
26                     op='-' ;
27                     break ;;
28                 multiply)
29                     op='*' ;
30                     break ;;
31                 divide)
32                     op='/' ;
33                     break ;;
34             *)
35                 echo "whyyyyyy:)"
36                 ;;
37             esac
38         done
39         ans=$(echo "$a $op $b" | bc -l)
40         printf "%s %s %s = %s\n\n" "$a" "$op" "$b" "$ans"
41     done
```

در این فایل دو عدد را از کاربر درخواست می کنیم و تا زمانی که عدد را به درستی وارد نکرده از او میخواهیم که دوباره ورودی را به درستی (یعنی عدد باشد) بدهد. سپس در case با توجه به گزینه ای که کاربر از میان operation ها select کرده است خروجی را به او تحویل می دهیم و ...

یک نمونه خروجی:

```
javad@javad-HP-350-G1: ~/Desktop
javad@javad-HP-350-G1:~/Desktop$ ./calculator.sh
Enter first number: 3
Enter second number: d5
please enter a number nothing else: 34
1) add
2) subtract
3) multiply
4) divide
choos operation please: 1
3 + 34 = 37

Enter first number: k1
please enter a number nothing else: kp
please enter a number nothing else: 54
Enter second number: 65
1) add
2) subtract
3) multiply
4) divide
choos operation please: 4
54 / 65 = .83076923076923076923

Enter first number: 
```

File_cat.sh

```
home > javad > Desktop > $ file_cat.sh > [0] counter
1  #!/bin/bash
2
3  x=$1
4  y=$2
5  file_name=$3
6
7  counter=1
8  let x=x-1
9  let y=y+1
10
11 while IFS= read -r line; do
12     if [[ $counter > $x ]]; then
13         if [[ $counter < $y ]]; then
14             let counter=counter+1
15             printf '%s\n' "$line"
16         fi
17     else
18         let counter=counter+1
19     fi
20 done < $file_name
21
```

در این فایل دو عدد و یک اسم فایل به عنوان آرگومان ورودی دریافت می کنیم و با خواندن تک تک خط های فایل و بررسی کردن اینکه آیا در بازه دلخواه هست یا نه و اینکه آنرا به خروجی چاپ کنیم اگر در بازه ورودی کاربر بود (یعنی خط های xام تا yام را چاپ کنیم)

نمونه:

```
javad@javad-HP-350-G1: ~/Desktop
javad@javad-HP-350-G1:~/Desktop$ bash file_cat.sh 2 5 test_file
line 2
line 3
line 4
line 5
javad@javad-HP-350-G1:~/Desktop$
```

Reverse.sh

در این فایل عدد ورودی کاربر را به صورت برعکس شده به آن بر می گردانیم و از تکنیک معروف تقسیم های متوالی به 10 استفاده می کنیم:

```
1  #!/bin/bash
2
3  read -r -p "Enter your number: " n
4  baghimande=0
5  rev_n=0
6  sum=0
7  while [ $n -gt 0 ]
8  do
9      baghimande=$(( $n % 10 ))
10     sum=$(( $sum + $baghimande ))
11     rev_n=$(( $rev_n * 10 + $baghimande ))
12     n=$(( $n / 10 ))
13 done
14 echo "revers: $rev_n"
15 echo "summation of digits: $sum"
```

```
javad@javad-HP-350-G1: ~/Desktop
javad@javad-HP-350-G1:~/Desktop$ ./reverse.sh
Enter your number: 4366340
revers: 436634
summation of digits: 26
javad@javad-HP-350-G1:~/Desktop$ ./reverse.sh
Enter your number: 4462
revers: 2644
summation of digits: 16
javad@javad-HP-350-G1:~/Desktop$
```

در این فایل با توجه به رودی کاربر یکی از 3 شکل موجود را به او نشان می دهیم و این کار را با استفاده از case انجام می دهیم. اما برای کشیدن شکل ها نیز می توان با استفاده از حلقه های for این کار را انجام داد.

```

1  #!/bin/bash
2
3  PS3="pick a number: "
4  select num in 1 2 3; do
5      case $num in
6          1)
7              for var in $(seq 1 5)
8              do
9                  for i in $(seq 1 $var)
10                 do
11                     printf "%d" $var
12                 done
13                 printf "\n"
14             done
15             break ;;
16          2)
17              for var in $(seq 1 6)
18              do
19                  let space=5-$var
20                  for var1 in $(seq 0 $space)
21                  do
22                      printf " "
23                  done
24                  for var2 in $(seq 1 $var)
25                  do
26                      printf "*"
27                  done
28                  printf "\n"
29             done
30
31             for var in $(seq 1 6)
32             do
33                 let var=7-$var
34                 let space=5-$var
35                 for var1 in $(seq 0 $space)
36                 do
37                     printf " "
38                 done
39                 for var2 in $(seq 1 $var)
40                 do
41                     printf "*"
42                 done
43                 printf "\n"
44             done

```

```

45
46         break ;;
47     3)
48         for var in $(seq 0 4)
49         do
50             for var1 in $(seq 1 $var)
51             do
52                 printf "| "
53             done
54             printf "|_\n"
55         done
56         break ;;
57     *)
58         echo "wrong input!!!"
59         ;;
60 esac
61 done

```

```

javad@javad-HP-350-G1:~/Desktop$ ./shekl.sh
1) 1
2) 2
3) 3
pick a number: 1
1
22
333
4444
55555
javad@javad-HP-350-G1:~/Desktop$ ./shekl.sh
1) 1
2) 2
3) 3
pick a number: 2
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
javad@javad-HP-350-G1:~/Desktop$ ./shekl.sh
1) 1
2) 2
3) 3
pick a number: 3
|_
| |
| | | | |
| | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | |
| | |
| |
|
javad@javad-HP-350-G1:~/Desktop$

```