

# به نام خدا

آزمایشگاه سیستم عامل

آزمایش ۳

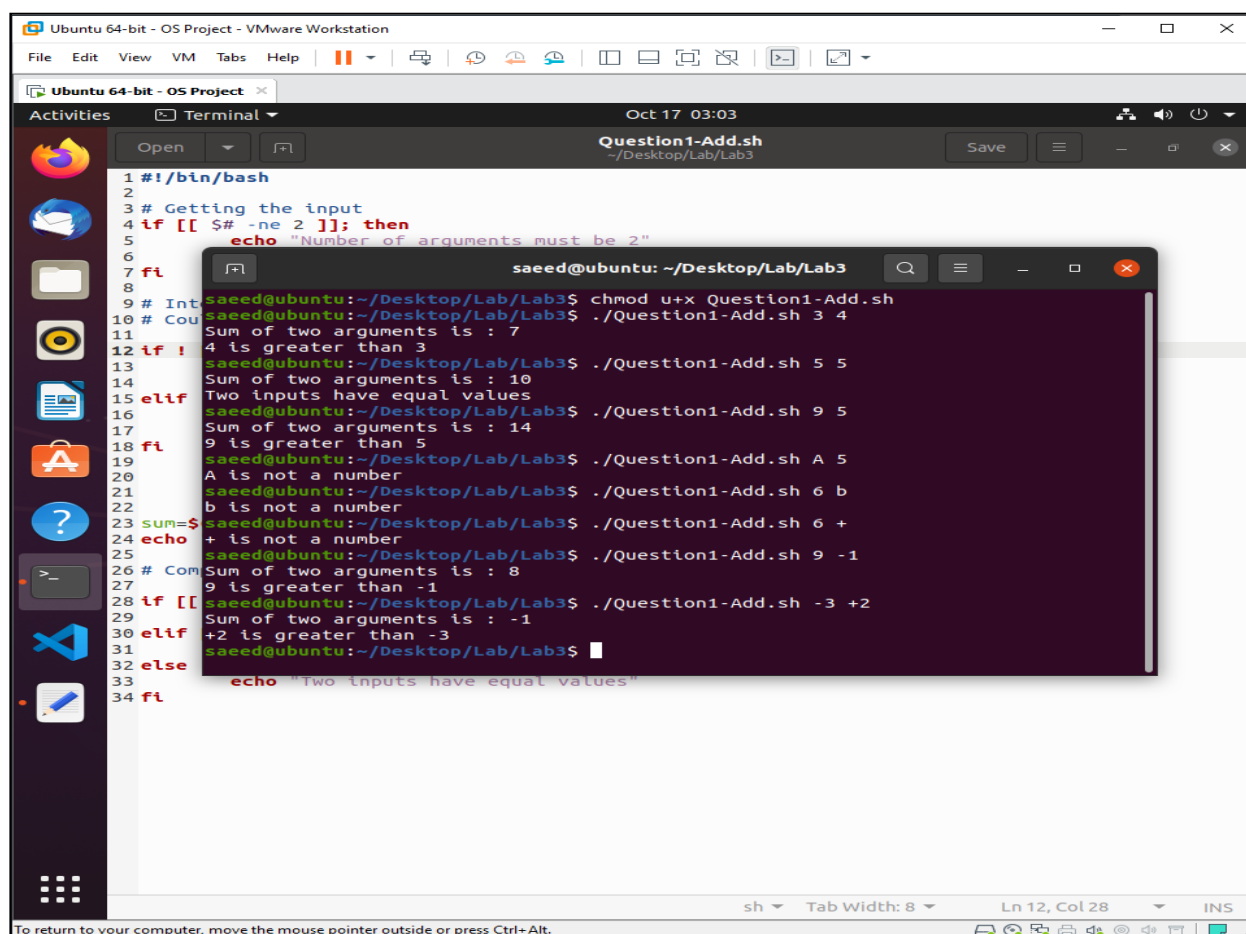
استاد: حسینی تودشکی

سیدسعید شفیعی

مهر ۱۴۰۰

## سوالات:

۱. در این سوال باید یک اسکریپ بنویسیم که دو عدد را جمع کند. برای بهتر عمل کردن برنامه باید دو شرط را بررسی کنیم. اینکه کاربر تنها ۲ عدد را به عنوان ورودی وارد کند و هم چنین ۲ متغیر وارد شده از نوع عدد باشند. پس از بررسی شرط ها ، دو عدد را جمع زده و در متغیر SUM می ریزیم و آن را نمایش می دهیم. در انتها هم بررسی می کنیم کدام یک از ۲ عدد، عدد بزرگتر است. برای هر ۳ حالتی که وجود دارد پیغام مناسب به کاربر نمایش می دهیم.
- برای اجرا از طریق bash شاید به ارور Permission Denied برخورد کنیم. با توجه به آزمایش قبلی می توان با دستور `chmod u+x "file_name"` دسترسی اجرا کردن فایل از طریق bash را داد ( برای تمام سوالات انجام شود).
- این اسکریپ قابلیت جمع کردن اعداد مثبت و منفی را دارد. پس به نوعی می توان این ماشین حساب را برای تمام اعداد حقیقی استفاده کرد ( بخش امتیازی با توجه به نظر استاد)



```
1 #!/bin/bash
2
3 # Getting the input
4 if [[ $# -ne 2 ]]; then
5     echo "Number of arguments must be 2"
6 fi
7
8 # Int
9 # Cou
10 # Cou
11 # Cou
12 if !
13     Sum of two arguments is : 7
14     4 is greater than 3
15     Sum of two arguments is : 10
16     Two inputs have equal values
17 elif
18     Sum of two arguments is : 14
19     9 is greater than 5
20 fi
21     A is not a number
22     b is not a number
23 sum=$
24 echo
25     Sum of two arguments is : 8
26 # Com
27     9 is greater than -1
28 if [[
29     Sum of two arguments is : -1
30 elif
31     +2 is greater than -3
32 else
33     echo "Two inputs have equal values"
34 fi
```

Terminal output:

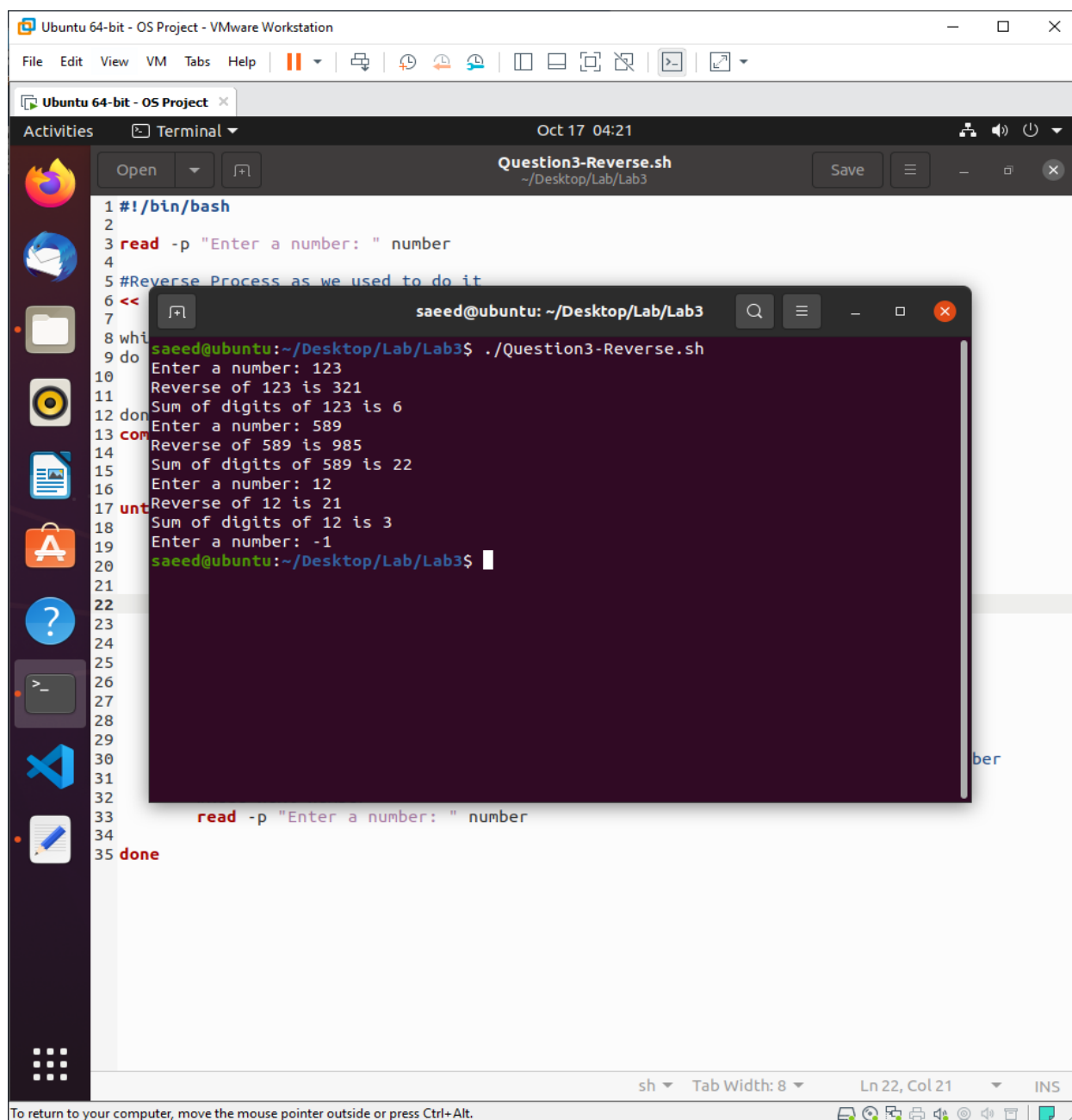
```
saeed@ubuntu: ~/Desktop/Lab/Lab3
saeed@ubuntu:~/Desktop/Lab/Lab3$ chmod u+x Question1-Add.sh
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 3 4
Sum of two arguments is : 7
4 is greater than 3
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 5 5
Sum of two arguments is : 10
Two inputs have equal values
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 9 5
Sum of two arguments is : 14
9 is greater than 5
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh A 5
A is not a number
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 6 b
b is not a number
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 6 +
+ is not a number
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh 9 -1
Sum of two arguments is : 8
9 is greater than -1
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question1-Add.sh -3 +2
Sum of two arguments is : -1
+2 is greater than -3
saeed@ubuntu:~/Desktop/Lab/Lab3$
```

۲. در این سوال باید یک ماشین حساب با استفاده از ساختار case طراحی کنیم. کاربر با وارد کردن دو عدد و Operand مربوط به عملی که می خواهد انجام دهد، نتیجه کار خود را دریافت می کند.

```
saeed@ubuntu: ~/Desktop/Lab/Lab3
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question2-Calculator.sh
Simple calculator
Insert 2 numbers and 1 operand , get your result
Enter first number:
5
Enter second number:
-2
Enter operand:
+
Result: 3
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question2-Calculator.sh
Simple calculator
Insert 2 numbers and 1 operand , get your result
Enter first number:
5
Enter second number:
-2
Enter operand:
-
Result: 7
saeed@ubuntu:~/Desktop/Lab/Lab3$
```

```
saeed@ubuntu: ~/Desktop/Lab/Lab3
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question2-Calculator.sh
Simple calculator
Insert 2 numbers and 1 operand , get your result
Enter first number:
4
Enter second number:
-3
Enter operand:
*
Result: -12
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question2-Calculator.sh
Simple calculator
Insert 2 numbers and 1 operand , get your result
Enter first number:
6
Enter second number:
-2
Enter operand:
/
Result: -3
saeed@ubuntu:~/Desktop/Lab/Lab3$
```

۳. در این سوال به طور متوالی عدد از کاربر دریافت می کنیم. راه اولی که به ذهن می رسد و بارها آن را با زبان های مختلف برنامه نویسی پیاده کرده ایم این است که با تقسیم های پیاپی بر ۱۰ می توان ارقام آن عدد را جدا جدا بدست آورد و حاصل جمع ارقام و عکس عدد را برای کاربر نمایش داد. برای بدست آوردن ارقام و حاصل جمع آنها، راه بهتری به ذهن نمی رسد اما برای برعکس کردن عدد می توان از تابع `rev` استفاده کرد. این تابع مقدار عکس خط های یک `text` را بر می گرداند و در اینجا ما می توانیم اعداد را هم `text` فرض کنیم و خروجی درست را دریافت کنیم. شرط پایان یافتن این تداوم را هم می توان وارد کردن عدد ۱- قرار داد.



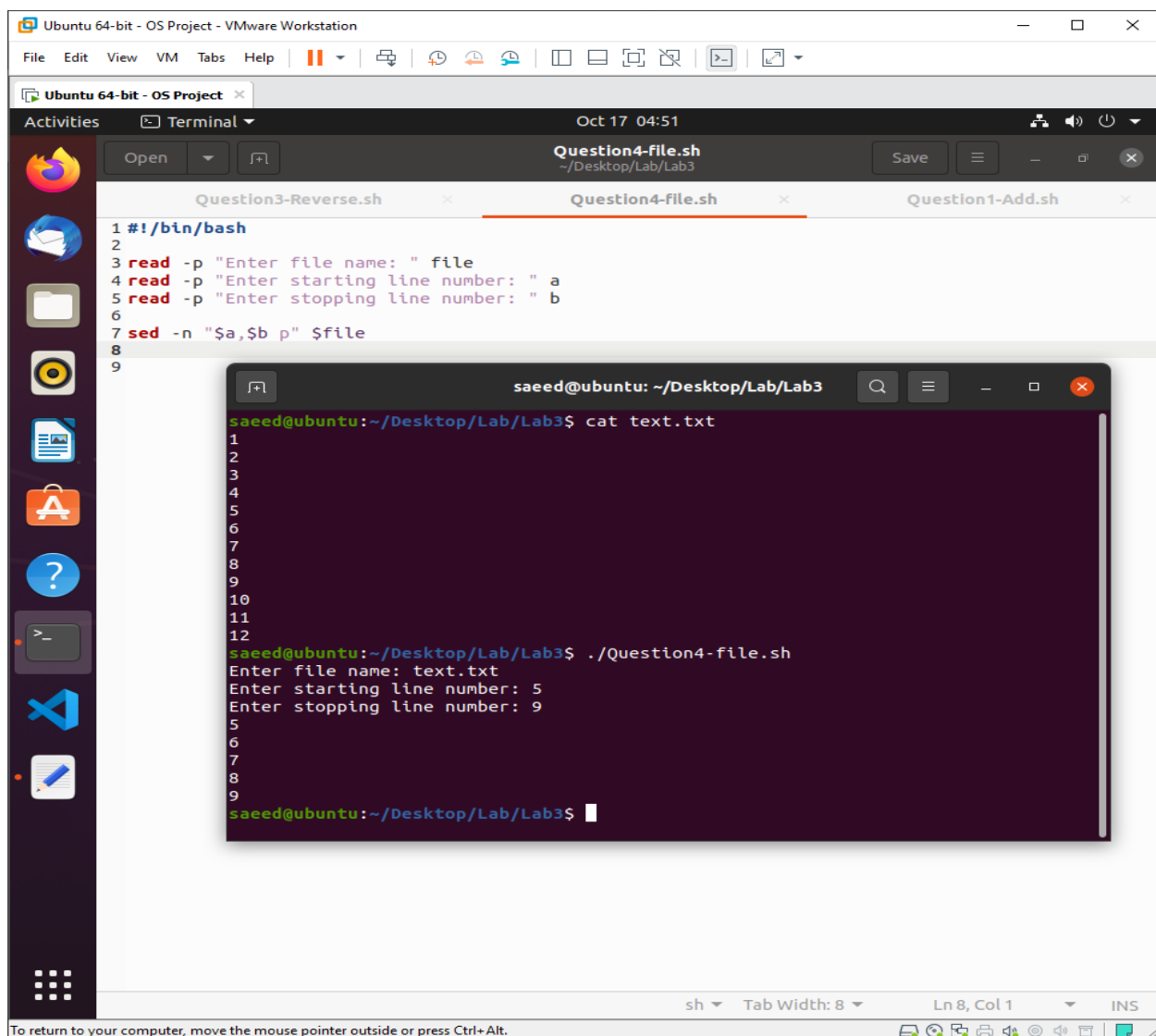
```

1 #!/bin/bash
2
3 read -p "Enter a number: " number
4
5 #Reverse Process as we used to do it
6 <<
7
8 while
9 do
10     Enter a number: 123
11     Reverse of 123 is 321
12     Sum of digits of 123 is 6
13 done
14 continue
15
16 Enter a number: 589
17 Reverse of 589 is 985
18 Sum of digits of 589 is 22
19 Enter a number: 12
20 Reverse of 12 is 21
21 Sum of digits of 12 is 3
22 Enter a number: -1
23 saeed@ubuntu:~/Desktop/Lab/Lab3$
24
25
26
27
28
29
30
31
32
33 read -p "Enter a number: " number
34
35 done
  
```

۴. در این سوال خواسته شده تا در یک فایل که نام آن ورودی داده می شود، در بازه ای که در ورودی داده می شود، خط های این فایل را چاپ کنیم. برای این کار راه های زیادی وجود دارد:

- استفاده از sed
- استفاده از NR , awk
- استفاده از دستورات head/tail
- استفاده از vi editor
- استفاده از perl

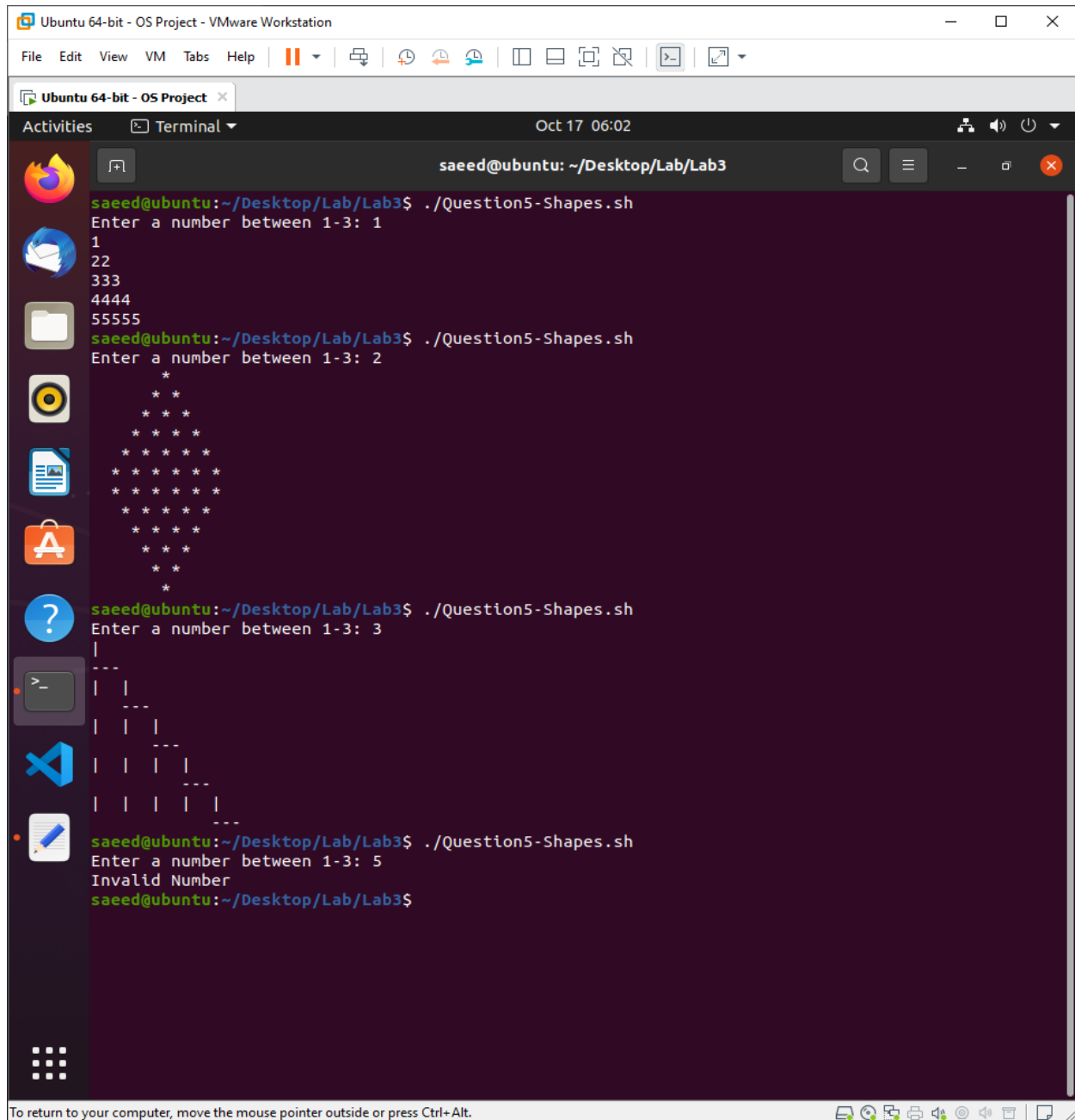
در این پیاده سازی از sed استفاده شده است.



```
#!/bin/bash
2
3 read -p "Enter file name: " file
4 read -p "Enter starting line number: " a
5 read -p "Enter stopping line number: " b
6
7 sed -n "$a,$b p" $file
8
9
```

```
saeed@ubuntu: ~/Desktop/Lab/Lab3
1
2
3
4
5
6
7
8
9
10
11
12
saeed@ubuntu:~/Desktop/Lab/Lab3$ cat text.txt
1
2
3
4
5
6
7
8
9
10
11
12
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question4-file.sh
Enter file name: text.txt
Enter starting line number: 5
Enter stopping line number: 9
5
6
7
8
9
saeed@ubuntu:~/Desktop/Lab/Lab3$
```

۵. در این سوال خواسته شده تا با دریافت عدد ۱ یا ۲ یا ۳، شکل متناظری که درون دستورکار ذکر شده است را رسم کنیم. برای هر کدام از اعداد متناظر، تابعی برای رسم شکل مناسب نوشته شده است. پس از دریافت ورودی و مطابقت آن با تابع مورد نظر، شکل در خروجی نمایش داده می شود. اگر عدد در بازه خواسته شده نبود، خروجی Invalid Number چاپ می شود.



```
Ubuntu 64-bit - OS Project - VMware Workstation
File Edit View VM Tabs Help
Ubuntu 64-bit - OS Project
Activities Terminal Oct 17 06:02
saeed@ubuntu: ~/Desktop/Lab/Lab3
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question5-Shapes.sh
Enter a number between 1-3: 1
1
22
333
4444
55555
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question5-Shapes.sh
Enter a number between 1-3: 2
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question5-Shapes.sh
Enter a number between 1-3: 3
|
| |
| | | | |
| | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | |
| | |
| |
|
saeed@ubuntu:~/Desktop/Lab/Lab3$ ./Question5-Shapes.sh
Enter a number between 1-3: 5
Invalid Number
saeed@ubuntu:~/Desktop/Lab/Lab3$
```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

## سوالات اضافه :

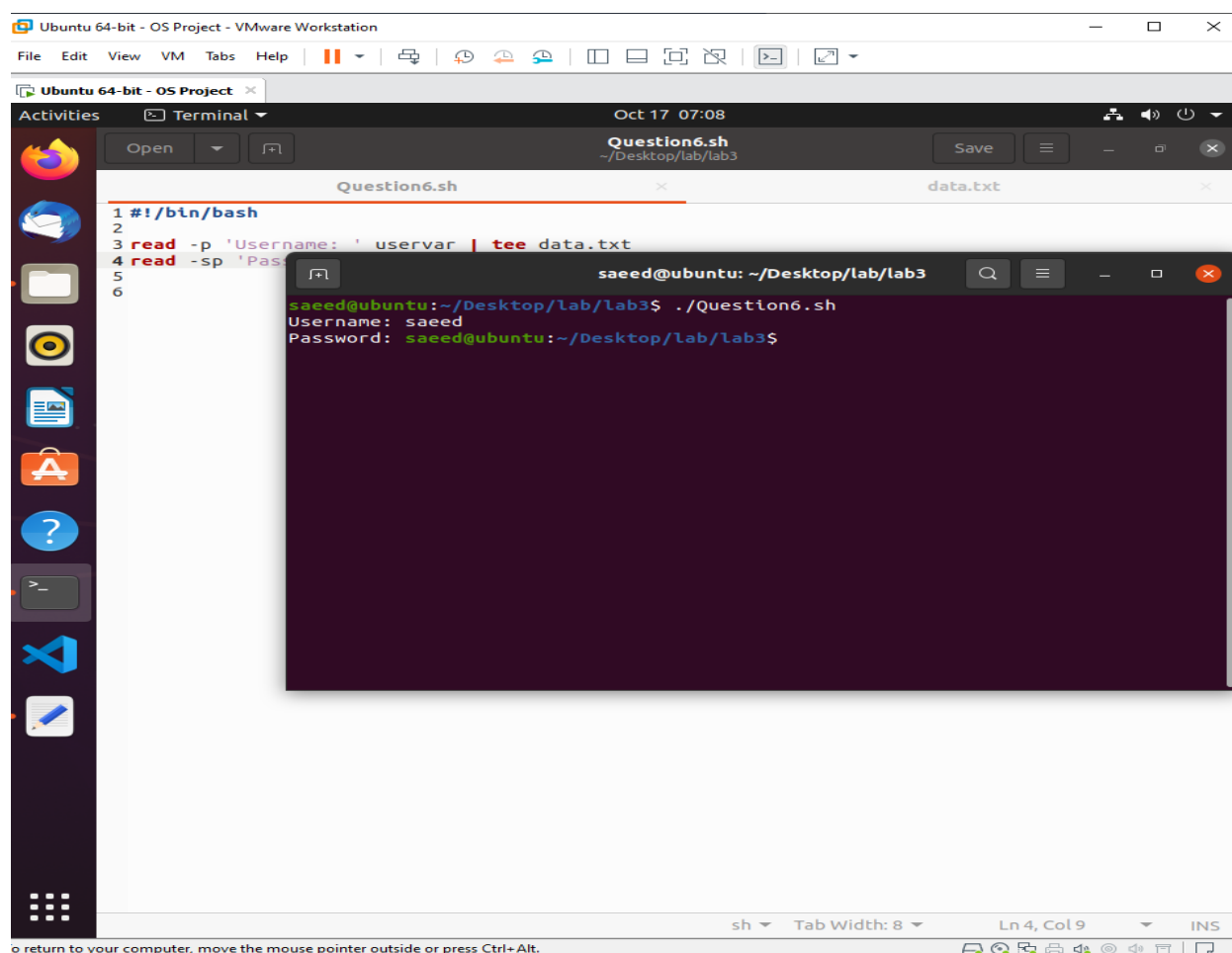
۱. برای دسترسی به دهمین آرگومان کافی است از {} استفاده کنیم. به عنوان مثال برای چاپ آرگومان دهم :

`echo "${10}"`

۲. سوییچ های `-p` و `-sp` :

به کمک `p` می توان یک رشته را پیش از ورودی گرفتن به عنوان راهنما به متن مورد نظر ورودی به کاربر نمایش داد.

به کمک `s` می توان کاراکتر های وارد شده در خروجی نوشته نشوند (مثلا برای وارد کردن رمز)

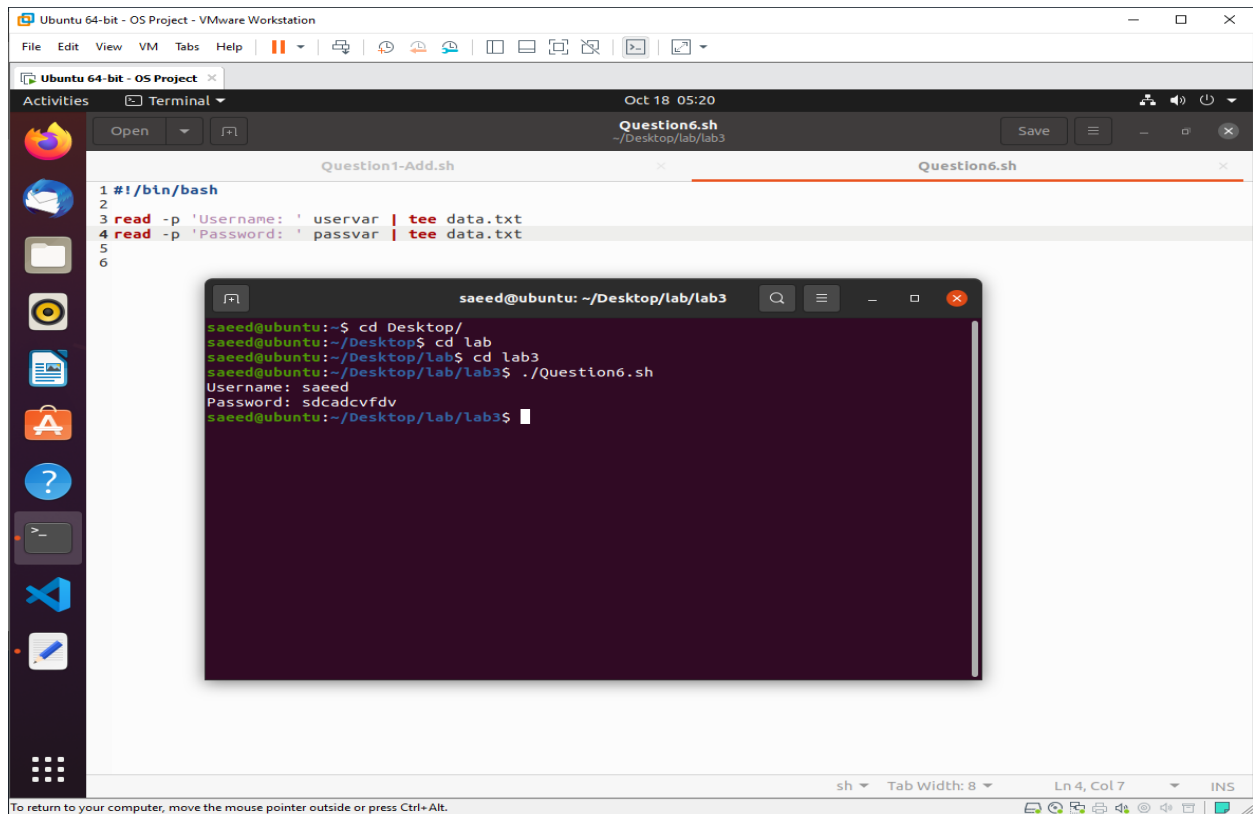


```
#!/bin/bash
2
3 read -p 'Username: ' uservar | tee data.txt
4 read -sp 'Pas
5
6
```

```
saeed@ubuntu: ~/Desktop/lab/lab3$ ./Question6.sh
Username: saeed
Password: saeed@ubuntu:~/Desktop/lab/lab3$
```

رمز وارد شده نمایش داده نشده است.

برای درک بهتر بدون کاراکتر S هم تست شده است که رمز نمایش داده می شود.



```
1 #!/bin/bash
2
3 read -p 'Username: ' uservar | tee data.txt
4 read -p 'Password: ' passvar | tee data.txt
5
6
```

```
saeed@ubuntu: ~/Desktop/lab/lab3
saeed@ubuntu:~$ cd Desktop/
saeed@ubuntu:~/Desktop$ cd lab
saeed@ubuntu:~/Desktop/lab$ cd lab3
saeed@ubuntu:~/Desktop/lab/lab3$ ./Question6.sh
Username: saeed
Password: sdcadcvcfdv
saeed@ubuntu:~/Desktop/lab/lab3$
```

۳. ماشین حساب مربوط به اعداد حقیقی درون خود سوال دوم که مربوط به ماشین حساب است پیاده شده است.