

گزارش کار آزمایش سوم

آرمین گودرزی-۹۸۳۱۰۵۵

سوال اول)
کد:

```
$ 1.sh
1  #!/bin/bash
2
3  case $1 in
4  | ''|*[!0-9]*) echo your first input is not an integer ;exit 1 ;
5  esac
6  case $2 in
7  | ''|*[!0-9]*) echo your second input is not an integer ;exit 1 ;
8  esac
9
10 sum=$(( $1+$2 ))
11
12 echo "Sum is: $sum"
13
14 if [ $1 -gt $2 ]
15 then
16     echo "greater number is $1";
17 elif [ $2 -gt $1 ]
18 then
19     echo "greater number is $2";
20 else
21     echo "numbers are even";
22 fi
```


توضیح کد)

الف) برای پیاده سازی این دستور تنها لازم بود که ۲ متغیر را با هم جمع کرده و در متغیر sum ریخته و در نهایت آن را چاپ کنیم.

ب) برای پیاده سازی این بخش نیز از دستورات شرطی استفاده کردیم که حالت آخر یعنی else برای حالت برابر بودن این دو عدد است.

ج) در ۹ خط اول برنامه چک میکنیم که هر کدام از ورودی های ما حتما عدد باشد و در غیر این صورت با پرینت کردن پیام مناسب از برنامه خارج میشویم.

نکته: بخش ج برای تجربه بهتر برای استفاده از برنامه در اول کد قرار داده شده تا ورودی ها را چک کند.

نمونه خروجی:

```
armin@ubuntu:~/Desktop/os-lab$ sh 1.sh 56 10
Sum is: 66
greater number is 56
armin@ubuntu:~/Desktop/os-lab$ sh 1.sh e 5
your first input is not an integer
armin@ubuntu:~/Desktop/os-lab$ █
```


سوال دوم)

کد:

```
$ 2.sh
1  #!/bin/bash
2
3  case $1 in
4  | ''|*[!0-9]*) echo your first input is not an integer ;exit 1 ;
5  esac
6  case $3 in
7  | ''|*[!0-9]*) echo your second input is not an integer ;exit 1 ;
8  esac
9
10
11 case $2 in
12 | '+')
13     echo $(( $1 + $3 ))
14     ;;
15 | '-')
16     echo $(( $1 - $3 ))
17     ;;
18 | 'x')
19     echo $(( $1 * $3 ))
20     ;;
21 | '/')
22     if [ $3 -eq 0 ]
23     then
24         echo "you cant devide a number by zero" ; exit 1
```


توضیح کد)

برای پیاده سازی ماشین حساب ابتدا باید چک کنیم که هر دو ورودی که به عنوان عدد میگیریم حتما عدد باشند که در ۸ خط اول کد چک میکنیم (روش مثل بخش ج سوال ۱ است) بعد روی متغیری که به عنوان عملگر گرفته ایم سویچ کیس را اجرا میکنیم البته برای جلوگیری از تقسیم بر ۰، برای عملگر تقسیم یک دستور شرطی هم داریم که چک میکنیم که عدد دوم ۰ نباشد.

نمونه خروجی:

```
armin@ubuntu:~/Desktop/os-lab$ sh 2.sh 3 / 0
you cant devide a number by zero
armin@ubuntu:~/Desktop/os-lab$ sh 2.sh 3 - 6
-3
armin@ubuntu:~/Desktop/os-lab$ sh 2.sh 3 - r
your second input is not an integer
armin@ubuntu:~/Desktop/os-lab$ █
```


سوال سوم)

کد:

```
$ 3.sh
1  #!/bin/bash
2
3  while :
4  do
5      echo "Enter your number: ";
6      read number;
7      reverse=0;
8      sumation=0;
9      while [ $number -gt 0 ]
10     do
11         reverse=$(( $reverse * 10 + $number % 10 ));
12         sumation=$(( $sumation + $number % 10 ));
13         number=$(( $number / 10 ));
14     done
15     echo 'reverse is: '$reverse ', sum is: ' $sumation;
16 done
```


توضیح کد)

برای پیاده سازی این سوال به ۲ حلقه تو در تو نیاز داریم. اولین حلقه برای اینکه مداوم عدد بگیریم و دومین حلقه نیز برای پیاده سازی عمل معکوس کردن عدد و جمع ارقام عدد میباشد.

در حلقه دوم تا زمانی که عدد بزرگتر از ۰ میباشد برای جمع ارقام آن را تقسیم بر ۱۰ میکنیم و برای معکوس کردن نیز در هر مرحله متغیر reverse را ضرب در ۱۰ کرده و با عدد تقسیم بر ۱۰ جمع میکنیم.

در نهایت نیز حاصل دو متغیر را پرینت کرده و عدد جدید را از کاربر میگیریم.

نمونه خروجی:

```
armin@ubuntu:~/Desktop/os-lab$ sh 3.sh
Enter your number:
456
reverse is: 654 , sum is:  15
Enter your number:
897
reverse is: 798 , sum is:  24
Enter your number:
232
reverse is: 232 , sum is:  7
Enter your number:
59067
reverse is: 76095 , sum is:  27
Enter your number:
4532
reverse is: 2354 , sum is:  14
Enter your number:
█
```

سوال چہارم)

کد:

```
$ 4.sh
2
3  echo "Enter x";
4  read x;
5
6  echo "Enter y";
7  read y;
8
9  echo "Enter file name";
10 read file_name;
11
12 sed -n $(( $x )), $(( $y ))p $file_name
```


توضیح کد)

۱۰ خط اول برنامه صرفاً گرفتن متغیرهای x و y و اسم فایل از کاربر میباشد.
در خط ۱۲ نیز با استفاده از دستور `sed` بازه x تا y از فایل داده شده را چاپ میکنیم.
(نکته) فایل استفاده شده در نمونه خروجی نیز در پوشه `codes` با نام `example` قرار دارد.

نمونه خروجی:

```
armin@ubuntu:~/Desktop/os-lab$ sh 4.sh
Enter x
2
Enter y
10
Enter file name
example
line2
line3
line4
line5
line6
line7
line8
line9
line10
armin@ubuntu:~/Desktop/os-lab$ █
```


سوال پنجم)

این سوال در واقع استفاده از تمام دستوراتی که تا به اینجا یاد گرفتیم به همراه استفاده از تابع را میطلبد. برای هرکدام از ۳ شکل داده شده یک تابع تعریف میکنیم که شکل اول و دوم ۲ حلقه تو در تو نیاز دارند و شکل سوم که پیچیده تر است اما ۲ حلقه که هرکدام درون خود ۲ حلقه دیگر نیز دارند را نیاز دارد. توابع این ۳ شکل با استفاده از حلقه for همینطور که درصدهای زیر میبینید از خط ۱ تا ۵۲ تعریف شده اند. در نهایت از خط ۵۴ تا آخر نیز از کاربر درخواست وارد کردن عدد ورودی را میکنیم و سپس با توجه به عدد یا یکی از شکل را با استفاده از تابع مورد نظر پرینت میکنیم یا پیغامی در مورد نا معتبر بودن عدد چاپ میکنیم.

```
$ 5.sh
1  #!/bin/bash
2
3  function print_shape1() {
4      for (( i=1; i<=5; i++ ));
5      do
6          for (( j=1; j<=$i; j++ ))
7          do
8              echo -n $i;
9          done
10         echo ' ';
11     done
12 }
13
14 function print_shape2() {
15     for (( i=1; i<=5; i++ ))
16     do
17         echo -n '|';
18         for (( j=2; j<=$i; j++ ))
19         do
20             echo -n ' |';
21         done
22         echo '_';
23     done
24 }
25
```


تابع شکل سوم

قسمت نهایی کد

```
53
54 echo -n "Enter number > ";
55 read number;
56
57 if [ $number = '1' ]
58 then
59     print_shape1
60 elif [ $number = '2' ]
61 then
62     print_shape2
63 elif [ $number = '3' ]
64 then
65     print_shape3
66 else
67     echo "invalid shape number";
68 fi
```

```
26 function print_shape3() {
27     for (( i=1; i<=5; i++ ))
28     do
29         for (( j=1; j<=$((5 - $i)); j++ ))
30         do
31             echo -n ' ';
32         done
33         for (( j=1; j <=$((2 * $i - 1)); j++ ))
34         do
35             echo -n '*';
36         done
37         echo ' ';
38     done
39     for (( i=1; i<=5; i++ ))
40     do
41         for (( j=1; j<=$i; j++ ))
42         do
43             echo -n ' ';
44         done
45         for (( j=1; j <=$((2 * (5 - $i) - 1)); j++ ))
46         do
47             echo -n '*';
48         done
49         echo ' ';
50     done
51 }
52
53
```


نمونہ خروجی

```
armin@ubuntu:~/Desktop/os-lab$ /bin/bash 5.sh
```

Enter number > 1

1

22

333

4444

55555

```
armin@ubuntu:~/Desktop/os-lab$ /bin/bash 5.sh
```

Enter number > 2

Number of children	Frequency
0	9
1	8
2	7
3	5
4	3
5	2
6	1

```
armin@ubuntu:~/Desktop/os-lab$ /bin/bash 5.sh
```

Enter number > 3

*

*

```
armin@ubuntu:~/Desktop/os-lab$ /bin/bash 5.sh
```

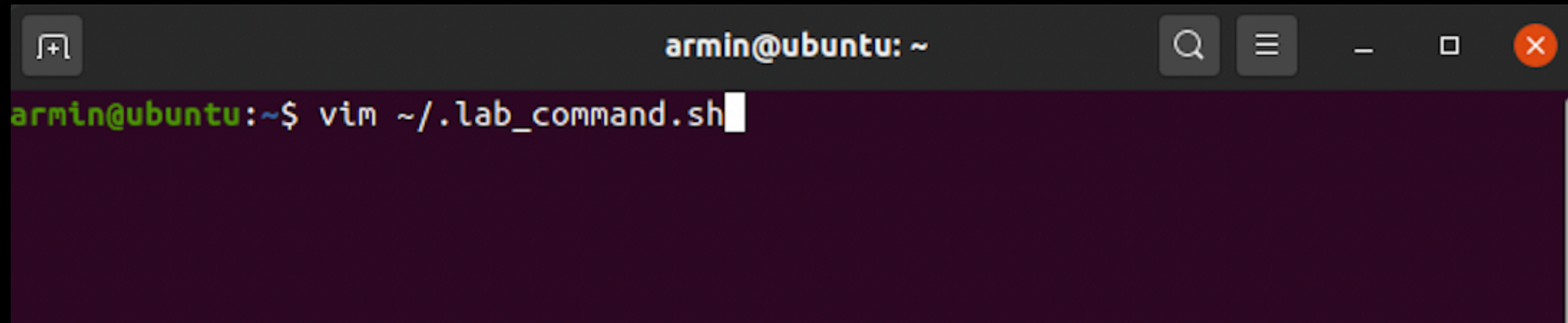
Enter number > 4

invalid shape number

```
armin@ubuntu:~/Desktop/os-lab$
```

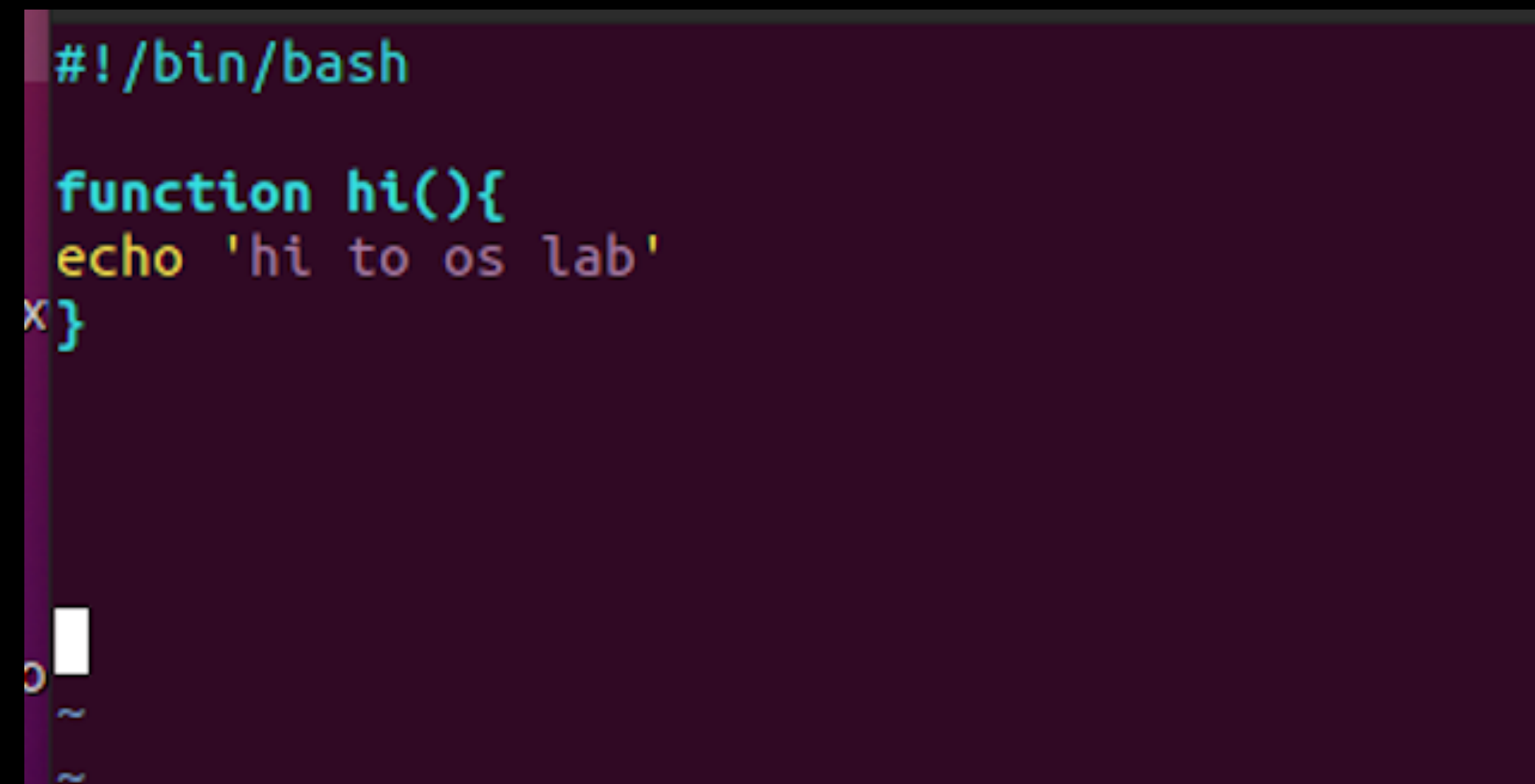

سوال ۶) اضافه کردن یک command به bash

ابتدا باید فایل bash خود را بنویسیم(در همان جایی که فایل bashrc قرار دارد یعنی root سیستم تا بعدا بتوانیم با مپ کردن آن هر جایی از سیستم command خود را صدا بزنیم)

A terminal window titled 'armin@ubuntu: ~' with standard window controls. The command 'vim ~/.lab_command.sh' is entered at the prompt.

```
armin@ubuntu:~$ vim ~/.lab_command.sh
```

فایل bash ما حاوی یک تابع است که یک پیام سلام چاپ می‌کند

A code editor window showing a bash script. The script defines a function 'hi()' that prints 'hi to os lab'.

```
#!/bin/bash

function hi(){
echo 'hi to os lab'
}
```


حال باید یک دستور به فایل `bashrc` اضافه کنیم تا دستور ما همه جا شناخته شده باشد پس ابتدا فایل `bashrc` را باز میکنیم

```
armin@ubuntu:~$ vim ~/.bashrc
```

دستور زیر به انتهای فایل اضافه میکنیم و ذخیره کرده و خارج میشویم

```
source ~/.lab_command.sh  
"~/.bashrc" 123L, 3825C
```

حال هرجایی از سیستم که یک ترمینال باز کنیم میتوانیم از تابع موجود در فایل `bash` ای که اضافه کردیم به عنوان یک `command` استفاده کنیم

```
armin@ubuntu: ~  
armin@ubuntu:~$ hi  
hi to os lab  
armin@ubuntu:~$
```