گزارش آزمایش سوم آزمایشگاه سیستم‌های عامل

اشکان شکیبا (۹۹۳۱۰۳۰)، علی هاشم‌پور (۹۹۳۱۰۸۲)

برنامه اول:

```
#!/bin/bash

echo "enter the first number"
read a
if [[ -n ${a//[0-9]/} ]]; then
    echo "not a number"
    exit 1
fi
echo "enter the second number"
read b
if [[ -n ${b//[0-9]/} ]]; then
    echo "not a number"
    exit 1
fi
let sum=a+b
echo "sum is $sum"
if [[ $a -gt $b ]]; then
    echo "$a is greater than $b"
elif [[ $a -lt $b ]]; then
    echo "$a is less than $b"
else
    echo "$a is equal to $b"
fi
```

ابتدا با چاپ فرمان مناسب، عدد ورودی کاربر را خوانده و شرط عددی بودن متغیر مربوطه را بررسی می‌کند. سپس با روندی مشابه عدد بعدی را دریافت کرده و جمع آنها را محاسبه و چاپ می‌کند. در انتها نیز با یک if دارای سه شرط، مقادیر عددها را مقایسه می‌کند.

برنامه دوم:

```
#!/bin/bash

while(true)
do
echo "enter the first number"
```

```
read a
echo "enter the operator"
read op
echo "enter the second number"
read b
case $op in
    +) let sum=a+b
        echo "sum is $sum"
        ;;
    -) let sum=a-b
        echo "sum is $sum"
        ;;
    /) let sum=a/b
        echo "sum is $sum"
        ;;
    x) let sum=a*b
        echo "sum is $sum"
        ;;
    *) echo "invalid operator"
        ;;
esac
done
```

ابتدا دو عملوند و یک عملگر از ورودی خوانده و سپس با بررسی عملوند، یکی از چهار عمل اصلی را انجام داده و حاصل آن را در sum ذخیره می‌کند و آن را در خروجی چاپ می‌کند.

برنامه سوم:

```
#!/bin/bash

while(true)
do
echo "enter the number"
read a
out=0
sum=0
while [ $a -gt 0 ]
do
    out=$(expr $out \* 10)
    digit=$(expr $a % 10)
    out=$(expr $out + $digit)
    a=$(expr $a / 10)
    sum=$(expr $sum + $digit)
done
echo "reverse is $out"
echo "sum of digits is $sum"
done
```

ابتدا عدد مورد نظر را از ورودی خوانده و سپس در یک حلقه، هر بار رقم یکان آن را با محاسبه باقیمانده تقسیمش بر ۱۰، جدا می‌کند. سپس رقم جداشده را به شکل وارونه به out که معادل وارون عدد است افزوده و آن را با sum که معادل حاصل ارقام عدد است جمع می‌کند. در انتها حاصل این دو متغیر را چاپ می‌کند.

برنامه چهارم:

```
#!/bin/bash

echo "enter name of file"
read file
echo "enter x"
read x
echo "enter y"
read y
tail -n +$x $file | head -n $(($y-$x+1))
```

ابتدا نام فایل و دو عدد مورد نظر برای ابتدا و انتهای آن را از ورودی می‌خواند و سپس با دستور tail و head، خطوط نامطلوب را از ابتدا و انتهای جدا می‌کند و حاصل را نمایش می‌دهد.

برنامه پنجم:

```
#!/bin/bash

echo "enter the number"
read a
case $a in
    1)
        for ((i=1; i<=5; i++)); do
            for ((j=1; j<=i; j++)); do
                echo -n $i
            done
            echo
        done
        echo
    ;;
    2)
```

```
        for ((i=0; i<6; i++)); do
            for ((j=0; j<5-i; j++)); do
                echo -n " "
            done
            for ((j=0;j<i+1;j++)); do
                echo -n "."
                echo -n " "
            done
            echo
        done
        for ((i=0; i<5; i++)); do
            for ((j=0; j<i+1; j++)); do
                echo -n " "
            done
            for ((j=0;j<5-i;j++)); do
                echo -n "."
                echo -n " "
            done
            echo
        done
    ;;
    3)
        for ((i=0;i<5;i++)); do
            echo -n "|"
            for ((j=0;j<i;j++)); do
                echo -n " |"
            done
            echo "_"
        done
    ;;
    *) echo "invalid number"
    ;;
esac
```

ابتدا عدد شکل را دریافت کرده و سپس بسته به مقدار آن، یکی از سه شکل را نمایش می‌دهد.

شکل اول با یک حلقه تو در تو برای ردیف‌ها و عددها چاپ می‌شود.

شکل دوم در دو حلقه تو در تو، نقاط را به شکل خواسته شده چاپ می‌کند.

شکل سوم نیز یک حلقه تو در تو دارد که علامات | و _ را به شکل مورد نظر نمایش می‌دهد.