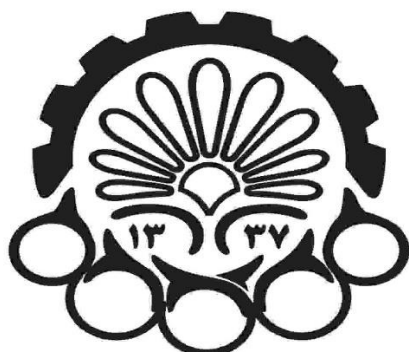


به نام خدا



**دانشگاه صنعتی امیر کبیر**  
( پلی تکنیک تهران )

دانشکده مهندسی کامپیوتر

آزمایشگاه سیستم های عامل

آزمایش دوم : آشنایی بادرستور نویسی درسیستم عامل

و خودکارسازی کارهای لازم در خط دستور

اعضای گروه :

محمد امین فرح بخش - (40131029)

حسین تاتار - (40133014)

اسفند 1403

**تمرین 1** ) کدی بنویسید که عددی را به عنوان ورودی آرگومان از کاربر دریافت کند. در صورتی که عدد کوچک تر از صفر بود، عبارت 'The weather is freezing' و در صورتی که بین صفر و 30 درجه بود، عبارت 'The weather is cool' و در صورتی که دما بالای 30 درجه بود، عبارت 'The weather is hot' را در ترمینال چاپ کند در این تمرین ابتدا فایل 1.sh را می سازیم که کدها را در آن بنویسیم . سه مثال می زنیم که تمام شرایط سوال را شامل شود .

```
amin@Frb: ~/Desktop
GNU nano 7.2                               ./1.sh
#!/bin/bash

read number

if [ $number -lt 0 ]; then
    echo -e "The weather is freezing\n"
elif [ $number -lt 30 ]; then
    echo -e "The weather is cool\n"
else
    echo -e "The weather is hot\n"
fi
```

```
amin@Frb: ~/Desktop
amin@Frb:~/Desktop$ ./1.sh
bash: ./1.sh: Permission denied
amin@Frb:~/Desktop$ chmod +x 1.sh
amin@Frb:~/Desktop$ ./1.sh
2
The weather is cool

amin@Frb:~/Desktop$ ./1.sh
150
The weather is hot

amin@Frb:~/Desktop$ ./1.sh
-50
The weather is freezing

amin@Frb:~/Desktop$
```

**تمرین 2** ماشین حسابی با استفاده از case طراحی کنید که به عنوان عدد اول، عدد دوم و علامت ریاضی علامتی که می خواهیم بین این دو انجام بشود را به عنوان ورودی دریافت و عنوان خروجی نتیجه را به ما نشان دهد، در صورتی که ورودی ها نامعتبر بود یا در انجام عملیات دچار مشکل شدیم نیز خروجی مرتبط به آن نمایش داده شود.

در این تمرین ابتدا عدد اول، عدد دوم و عملگر را با استفاده از دستور read به عنوان ورودی دریافت می کنیم. در ادامه با استفاده از Case بررسی می کنیم که کدام عملگر اجرا شود. در صورتی عملیات نامعتبر باشد خروجی مناسب نمایش داده میشود .

```
amin@Frb: ~/Desktop
GNU nano 7.2                               ./2.sh
#!/bin/bash

echo "عدد اول را وارد کنید: "
read num1
echo "عدد دوم را وارد کنید: "
read num2
echo "عملگر (+ - * /): "
read operator

if ! [[ "$num1" =~ ^[0-9]+$ ]] || ! [[ "$num2" =~ ^[0-9]+$ ]]; then
    echo "خطا: لطفاً فقط عدد وارد کنید"
    exit 1
fi

case $operator in
    "+")
        result=$((num1 + num2))
        echo "نتیجه: $num1 + $num2 = $result"
        ;;
    "-")
        result=$((num1 - num2))
        echo "نتیجه: $num1 - $num2 = $result"
        ;;
    "*")
```

GNU nano 7.2

./2.sh

```

result=$((num1 + num2))
echo "نتیجه: $num1 + $num2 = $result"
;;
"-")
result=$((num1 - num2))
echo "نتیجه: $num1 - $num2 = $result"
;;
"*)
result=$((num1 * num2))
echo "نتیجه: $num1 * $num2 = $result"
;;
"/")
if [ $num2 -eq 0 ]; then
    echo "خطا: نمی توان برصفر تقسیم کرد"
else
    result=$((num1 / num2))
    echo "نتیجه: $num1 ÷ $num2 = $result"
fi
;;
*)
echo "خطا: عملگر نامعتبر! فقط از + - * / استفاده کنید"
;;
esac
echo $d

```

```
amin@Frb:~/Desktop$ chmod +x 2.sh
```

```
amin@Frb:~/Desktop$ ./2.sh
```

```
عدد اول را وارد کنید:
```

```
5
```

```
عدد دوم را وارد کنید:
```

```
9
```

```
عملگر (+ - * /):
```

```
*
```

```
نتیجه: 45 = 9 × 5
```

```
amin@Frb:~/Desktop$ 9
```

```
9: command not found
```

```
amin@Frb:~/Desktop$ ./2.sh
```

```
عدد اول را وارد کنید:
```

```
8
```

```
عدد دوم را وارد کنید:
```

```
0
```

```
عملگر (+ - * /):
```

```
/
```

```
خطا: نمی توان برصفر تقسیم کرد
```

```
amin@Frb:~/Desktop$ ./2.sh
```

```
عدد اول را وارد کنید:
```

```
p
```

```
عدد دوم را وارد کنید:
```

```
l
```

```
عملگر (+ - * /):
```

```
*
```

```
خطا: لطفاً فقط عدد وارد کنید
```

```
amin@Frb:~/Desktop$
```

**تمرین 3** در ادامه تمرین اول اسکریپ اولیه را به گونه ای تکمیل کنید که دمای سانتی گراد وارد شده توسط کاربر را به فارنهایت تبدیل کند و در خروجی نمایش دهد .

```
amin@Frb: ~/Desktop
GNU nano 7.2                               ./3.sh
#!/bin/bash

if [ $# -ne 1 ]; then
    exit 1
fi

number="$1"

if ! [[ "$number" =~ ^-?[0-9]+$ ]]; then
    exit 1
fi

if [ "$number" -lt 0 ]; then
    echo -e "The weather is freezing"
elif [ "$number" -lt 30 ]; then
    echo -e "The weather is cool"
else
    echo -e "The weather is hot"
fi

fahrenheit=$(echo "scale=2; $number * 9/5 + 32" | bc)
echo "دمای به فارنهایت: $fahrenheit°F"
```

```
amin@Frb: ~/Desktop
amin@Frb:~/Desktop$ chmod +x 3.sh
amin@Frb:~/Desktop$ ./3.sh 55
The weather is hot
131.00 °F دمای به فارنهایت:
amin@Frb:~/Desktop$ ./3.sh -80
The weather is freezing
112.00 °F دمای به فارنهایت:
amin@Frb:~/Desktop$ ./3.sh 200
The weather is hot
392.00 °F دمای به فارنهایت:
amin@Frb:~/Desktop$ ./3.sh 2
The weather is cool
35.60 °F دمای به فارنهایت:
amin@Frb:~/Desktop$ ./3.sh 0
The weather is cool
32.00 °F دمای به فارنهایت:
amin@Frb:~/Desktop$
```

**تمرین 4** ) کدی بنویسید که ارقام عددی را به شکل متوالی از کاربر دریافت کند، و عددی را چاپ کند که ترتیب ارقامش معکوس عدد اولی باشد. برای مثال در صورت دریافت ارقام به شکل 678 عدد 876 را در ترمینال چاپ کند .

```
amin@Frb: ~/Desktop
GNU nano 7.2                               ./4.sh
#!/bin/bash

read number

while [ $number -gt 0 ];
do
echo -n $(( number%10 ))
let number=$number/10
done

echo -e "\n"
```

```
amin@Frb: ~/Desktop
amin@Frb:~/Desktop$ chmod +x 4.sh
amin@Frb:~/Desktop$ ./4.sh
8956
6598

amin@Frb:~/Desktop$ ./4.sh
1234
4321

amin@Frb:~/Desktop$ 1111
1111: command not found
amin@Frb:~/Desktop$ ./4.sh
1111
1111

amin@Frb:~/Desktop$
```

## تمرین های امتیازی :

**تمرین 5** ) کدی بنویسید که آدرس مطلق یا نسبی یک دایرکتوری را به عنوان آرگومان دریافت میکند و به عنوان خروجی تعداد فایل های موجود در آن دایرکتوری را نمایش میدهد .

```
amin@Frb: ~/Desktop
GNU nano 7.2                               ./5.sh
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "لطفاً به آس دایرکتوی به عنوان آرگومان وارد کنید"
    exit 1
fi

directory="$1"

if [ ! -d "$directory" ]; then
    echo "یا وجود نداره یا به دایرکتوی نیست '$directory' :خطا"
    exit 1
fi

file_count=$(ls -l "$directory" | grep ^- | wc -l)

echo "تعداد فایلها در '$directory': $file_count"
```

```
amin@Frb: ~/Desktop
amin@Frb:~/Desktop$ ./5.sh .
تعداد فایلها در '.' : 5
amin@Frb:~/Desktop$ ls
1.sh 2.sh 3.sh 4.sh 5.sh
amin@Frb:~/Desktop$
```

همانطور که مشاهده میشود تعداد فایل های موجود در دایرکتوری ای که در آن قرار داریم 5 است .

حالا یک آدرس دیگر را امتحان میکنیم :

```
amin@Frb: ~  
amin@Frb:~/Desktop$ chmod +x 5.sh  
amin@Frb:~/Desktop$ ./5.sh  
خطا: لطفاً به آدرس دایرکتوری به عنوان آرگومان وارد کنید  
amin@Frb:~/Desktop$ ./5.sh ..  
تعداد فایل‌ها در '..': 4  
amin@Frb:~/Desktop$ cd ..  
amin@Frb:~$ ls  
1.sh Documents photo_2024-09-28_20-00-57.jpg ssh_to_server.sh  
agha_amin Downloads Pictures Templates  
Applications edit.txt Public Videos  
Desktop Music snap
```

مشاهده میشود که به درستی تعداد فایل‌ها را می‌شمارد. حال اگر فایل 1.sh را حذف کنیم یا جا به جا کنیم و دوباره امتحان کنیم :

```
amin@Frb: ~  
amin@Frb:~/Desktop$ ./5.sh ..  
تعداد فایل‌ها در '..': 3  
amin@Frb:~/Desktop$ cd ..  
amin@Frb:~$ ls  
agha_amin Downloads Pictures Templates  
Applications edit.txt Public Videos  
Desktop Music snap  
Documents photo_2024-09-28_20-00-57.jpg ssh_to_server.sh  
amin@Frb:~$
```

به درستی تعداد 3 را به ما نشان داد.

**سوال 6)** در رابطه با دستور awk مطالعه کنید و یک نمونه از نحوه کار با این دستور و استفاده از در تغییر دادن فایل‌های متنی با رشته‌های طولانی را به دلخواه نشان دهید.

دستور awk، ابزاری قدرتمند برای پردازش متن در سیستم‌های Linux و Unix است که عموماً برای پردازش و اسکن کردن الگوها مورد استفاده قرار می‌گیرد. به ما این امکان را میدهد تا داده‌ها را درون فایل‌های متنی تغییر دهیم و یا از آنها استخراج کنیم.



نحوه کار این دستور بدین صورت است که با خواندن خط به خط ورودی، قوانین مشخص شده را به هر خط اعمال می کند و عمل هایی مانند جست و جو، جایگزینی و فرمت کردن متن را انجام میدهد. به طور عمده از این دستور برای فیلتر کردن داده، ساختن گزارش روی آن و تغییر داده های ساختار مند مانند فایل های CSV استفاده می شود.

در ادامه 4 دستور awk را روی فایل edit.txt اجرا میکنیم :

```
Open  edit.txt
1. :cover="record?.id ? cover : page.cover"
2.dskfjodsfjgdhoifgodfhodhfiuggifufduiodfuioguioiudfuioggd
fiewhrewryew89ryew8yrew
4.jhvjkdksdvksdbvjkbkjbsvbsbdvdsnm
kljvljsdfw8reewjfkjfsdg212fefer
```

**1- چاپ خطوطی که با عدد شروع میشوند :**

```
amin@Frb: ~
amin@Frb:~$ awk '/^[0-9]/ {print $0}' edit.txt
1. :cover="record?.id ? cover : page.cover"
2.dskfjodsfjgdhoifgodfhodhfiuggifufduiodfuioguioiudfuioggd
4.jhvjkdksdvksdbvjkbkjbsvbsbdvdsnm
amin@Frb:~$
```

**2- استخراج فقط بخش قبل از نقطه : ( خطوطی که نقطه ندارند را کامل نمایش میدهد )**

```
amin@Frb:~$ awk -F'.' '{print $1}' edit.txt
1
2
fiewhrewryew89ryew8yrew
4
kljvljsdfw8reewjfkjfsdg212fefer
amin@Frb:~$
```

### 3- جایگزین کردن نقطه با خط تیره و چاپ :

```
amin@Frb:~$ awk '{gsub(/\./, "-"); print $0}' edit.txt
1- :cover="record?-id ? cover : page-cover"
2-dskfjodsfjgdhoifgodfhodhfiuggifufduiodfuioguioidfuioggd
fewhrewryew89ryew8yrew
4-jhvjkdksvksdbvjkbkjbsvbsbdvdsnm
kljvljsdfw8reewjfkjfsdg212fefer

amin@Frb:~$ █
```

### 4- شمارش تعداد کارکترهای هر خط :

```
amin@Frb:~$ awk '{print "Line " NR ": " length($0) " characters"}' edit.txt
Line 1: 43 characters
Line 2: 58 characters
Line 3: 24 characters
Line 4: 34 characters
Line 5: 31 characters
Line 6: 0 characters
amin@Frb:~$
```