

Linux və Shell Script

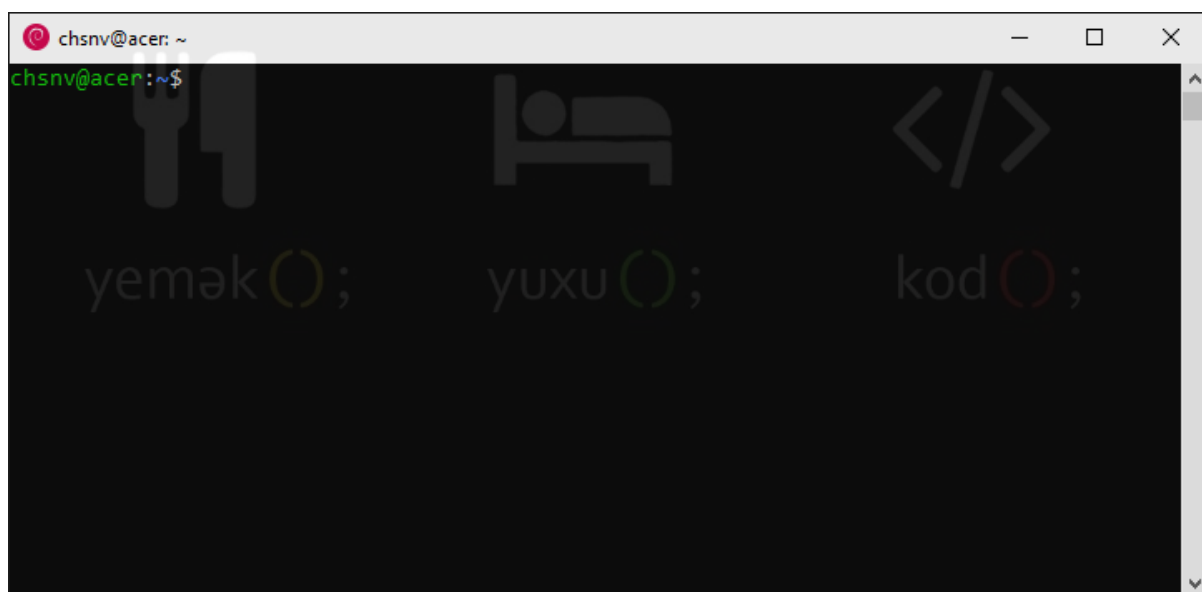
Linux'a giriş

Linux nədir?

Linux, ən sadə dildə desək, bir əməliyyat sistemidir. Tətbiqlərin və kompüter operatorunun kompüterdəki cihazlara istədiyi funksiyaları yerinə yetirməsini təmin edən kompüter proqramı. Linus Torvalds tərəfindən kernel yəni özəyi kodlanan və daha sonra istifadəçilər tərəfindən açıq qaynaq olması ilə inkişaf etdirilən əməliyyat sistemi. Linux açıq mənbəli "open source" bir əməliyyat sistemidir yəni ki, hər bir kəsin, istədiyi yerdə, istədiyi zaman düzəliş etmək və ehtiyaclarına uyğunlaşdırmaq üçün kodu ala biləcəyi mənasını verir.

Nədir Linux Terminal?

Linux terminalı CLI "Command Line Interface" yəni əmr sətiri interfeysi vasitəsilə fayl(ları), qovluq(ları), qurğu(ları), proqram(ları) idarə etmək və konfigurasiya etmək üçün bir "gate" yəni qapıdır.

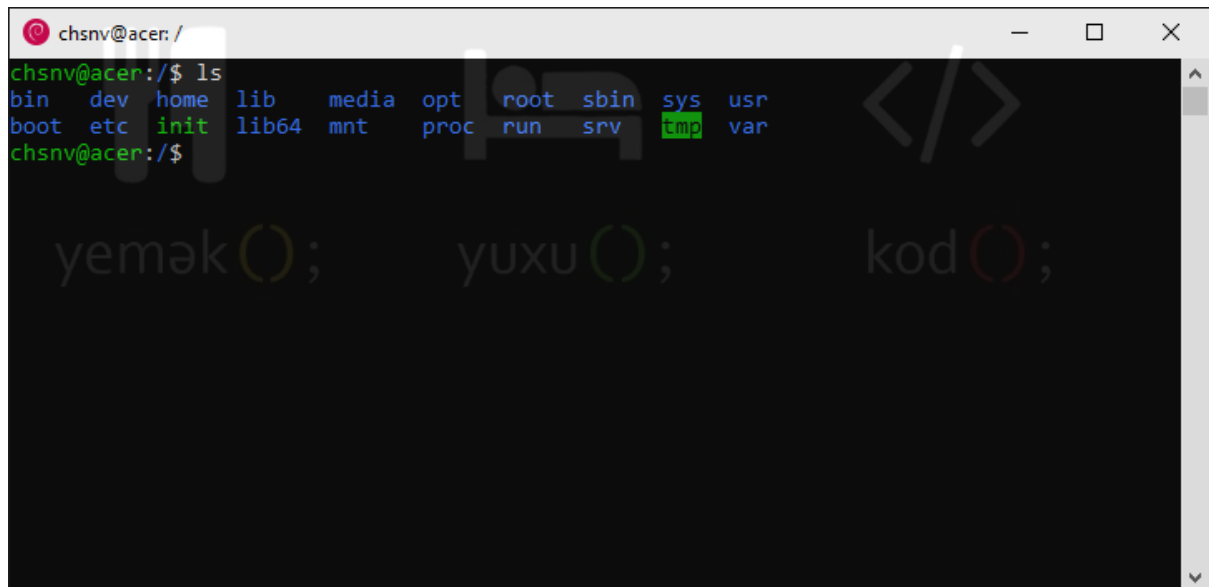


Bəs nə üçün Linux Terminal?

Niyə bəs yaxşı yeni GUI "Graphical User Interface" əvəzinə köhnə CLI istifadə etməliyik? Cavab isə səmərəlilik və gücdür. Terminal, sadə ssenarilərlə tapşırıqları avtomatlaşdırmağa imkan verir. Əks təqdirdə bir GUI istifadə edərək hər dəfə tapşırıqları "manual" etməli oluruq. Tipik istifadəçinin qarışdırması və ya buraxılmaması lazım olan Terminal adətən ən güclü əmrlərə malikdir.

Linux fayl sistemi iyerarxiyası

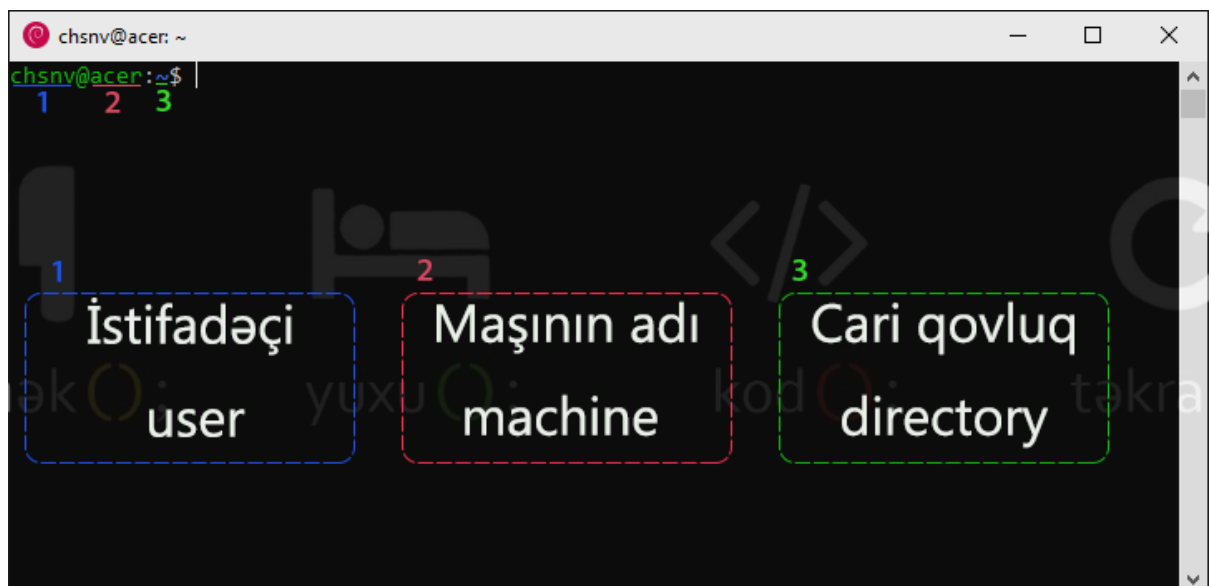
"/" - root yəni kök və alt direktoriyalrı



```
chsnv@acer: /  
chsnv@acer:/$ ls  
bin    dev    home  lib    media  opt    root  sbin  sys  usr  
boot   etc    init  lib64  mnt    proc   run   srv   tmp  var  
chsnv@acer:/$
```

Terminal əsasları

İnterfeys olaraq terminalın malik olduğu xüsusiyyətləri.



```
chsnv@acer: ~$
```

1 2 3

1 İstifadəçi
user

2 Maşının adı
machine

3 Cari qovluq
directory

Qeyd: 3. "~" yəni tilda /home/chsnv istifadəçi direktoriyasının qısa işarəsi.

1. Hazırda cari daxil olan istifadəçi.
2. Daxil olduğu maşın yəni qurğu.
3. Hazırda cari direktoriya.

Linux CLI

--help bu əmr mümkün olan seçimlərin siyahısı, bəzən necə istifadə ediləcəyi barədə bir təlimat verilir. Terminalda çıxışda yardım səhifəsini çap edir.

```
chsnv@acer: ~  
chsnv@acer:~$ mkdir --help  
Usage: mkdir [OPTION]... DIRECTORY...  
Create the DIRECTORY(ies), if they do not already exist.  
  
Mandatory arguments to long options are mandatory for short options too.  
-m, --mode=MODE    set file mode (as in chmod), not a=rwx - umask  
-p, --parents       no error if existing, make parent directories as needed  
-v, --verbose       print a message for each created directory  
-Z                 set SELinux security context of each created directory  
                    to the default type  
--context[=CTX]    like -Z, or if CTX is specified then set the SELinux  
                    or SMACK security context to CTX  
--help             display this help and exit  
--version          output version information and exit  
  
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>  
Full documentation at: <https://www.gnu.org/software/coreutils/mkdir>  
or available locally via: info '(coreutils) mkdir invocation'  
chsnv@acer:~$
```

Qeyd: Alternativ olaraq *man(ual)* əmri istifadə edilə bilər.

pwd

pwd - "print working directory" hazırkı mövcud direktoriya.

```
chsnv@acer: ~  
chsnv@acer:~$ pwd  
/home/chsnv  
chsnv@acer:~$
```

cd

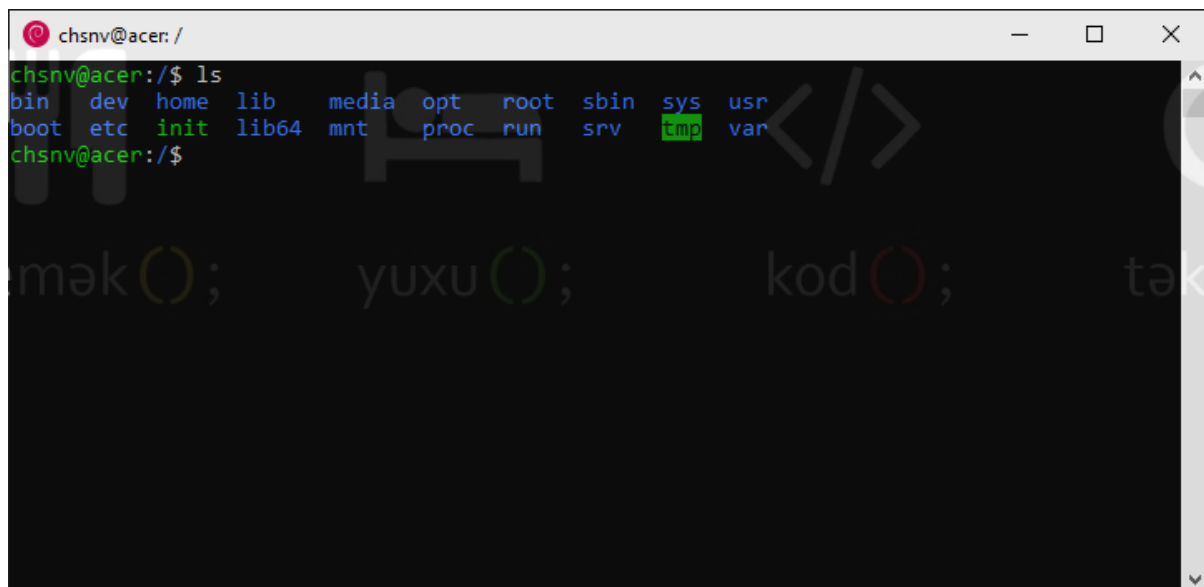
cd - "change directory" cari, mövcud direktoriyanı dəyişmək.

A terminal window titled 'chsnv@acer: /' with standard window controls. The command history shows: 'chsnv@acer:~\$ pwd' followed by '/home/chsnv', then 'chsnv@acer:~\$ cd /', then 'chsnv@acer:/\$ pwd' followed by '/', and finally 'chsnv@acer:/\$'.

```
chsnv@acer:~$ pwd
/home/chsnv
chsnv@acer:~$ cd /
chsnv@acer:/$ pwd
/
chsnv@acer:/$
```

ls

ls - "list" hazırki direktoriyanın fayl(ların) və qovluq(ların) siyahısı.

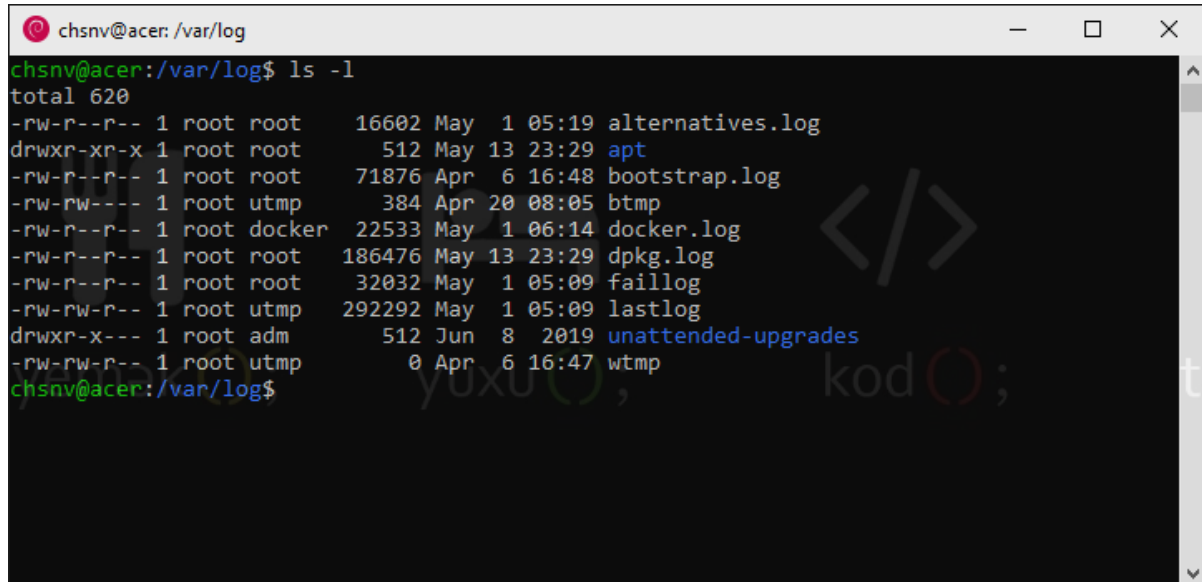
A terminal window titled 'chsnv@acer: /' with standard window controls. The command history shows: 'chsnv@acer:/\$ ls' followed by a multi-line listing of system directories. The 'tmp' directory is highlighted in green in the original image.

```
chsnv@acer:/$ ls
bin      dev      home     lib      media    opt      root     sbin     sys      usr
boot     etc      init     lib64    mnt      proc     run      srv      tmp      var
chsnv@acer:/$
```

Qeyd: Hər bir direktoriyanın özünə məxsus xüsusiyyətləri və mədsəqləri vardır misal kimi *home* direktoriyası istifadəçi(lərin) fayllarının saxlandığı yer, digərləri *var mnt temp etc opt* yəni optional kimi direktoriyaların öz xüsusiyyəti və mahiyyəti vardır.

ls -l

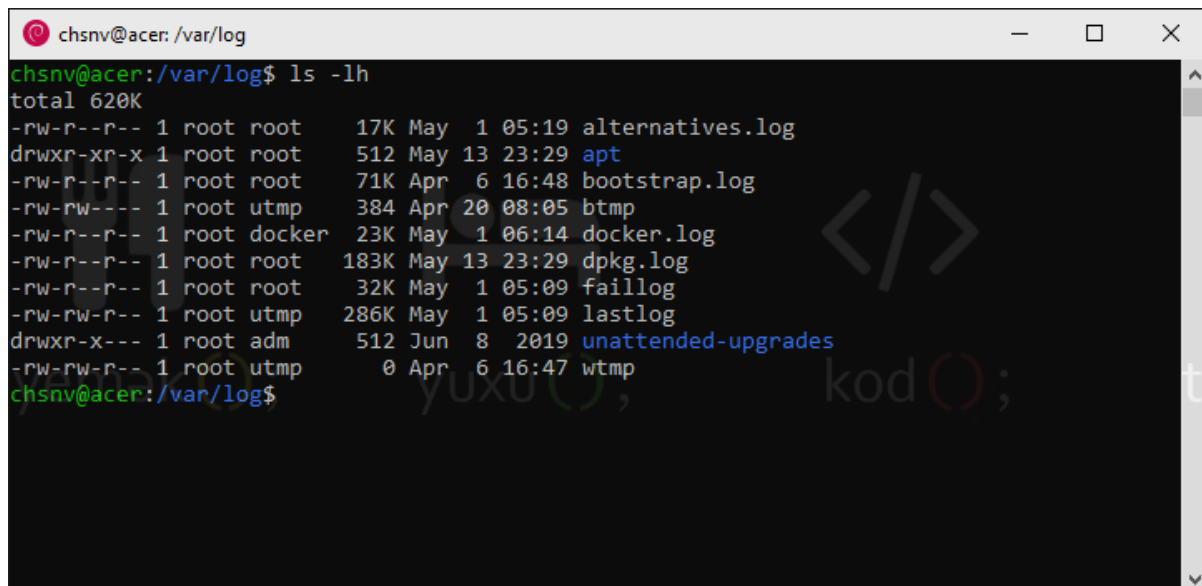
ls -l - direktoriyanın detalları ilə birgə siyahısı.



```
chsnv@acer: /var/log
chsnv@acer:/var/log$ ls -l
total 620
-rw-r--r-- 1 root root    16602 May  1 05:19 alternatives.log
drwxr-xr-x 1 root root      512 May 13 23:29 apt
-rw-r--r-- 1 root root   71876 Apr  6 16:48 bootstrap.log
-rw-rw---- 1 root utmp     384 Apr 20 08:05 btmp
-rw-r--r-- 1 root docker 22533 May  1 06:14 docker.log
-rw-r--r-- 1 root root 186476 May 13 23:29 dpkg.log
-rw-r--r-- 1 root root   32032 May  1 05:09 faillog
-rw-rw-r-- 1 root utmp 292292 May  1 05:09 lastlog
drwxr-x--- 1 root adm      512 Jun  8 2019 unattended-upgrades
-rw-rw-r-- 1 root utmp      0 Apr  6 16:47 wtmp
chsnv@acer:/var/log$
```

ls -lh

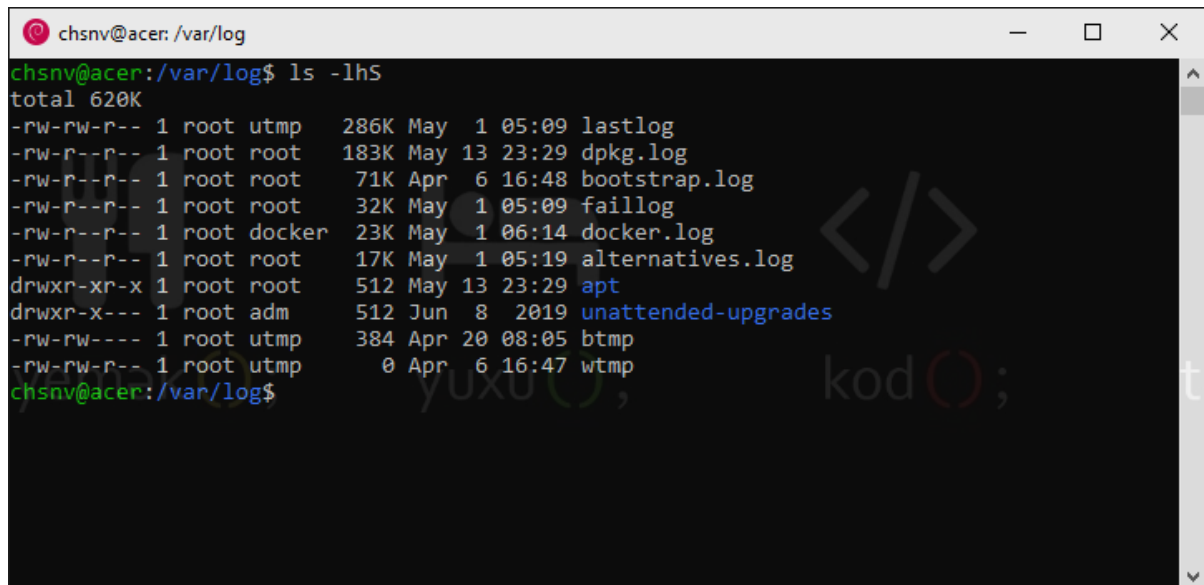
ls -lh - direktoriyanın insan oxuyacaq formatda detalları ilə birgə siyahısı.



```
chsnv@acer: /var/log
chsnv@acer:/var/log$ ls -lh
total 620K
-rw-r--r-- 1 root root    17K May  1 05:19 alternatives.log
drwxr-xr-x 1 root root     512 May 13 23:29 apt
-rw-r--r-- 1 root root    71K Apr  6 16:48 bootstrap.log
-rw-rw---- 1 root utmp     384 Apr 20 08:05 btmp
-rw-r--r-- 1 root docker  23K May  1 06:14 docker.log
-rw-r--r-- 1 root root   183K May 13 23:29 dpkg.log
-rw-r--r-- 1 root root    32K May  1 05:09 faillog
-rw-rw-r-- 1 root utmp   286K May  1 05:09 lastlog
drwxr-x--- 1 root adm      512 Jun  8 2019 unattended-upgrades
-rw-rw-r-- 1 root utmp      0 Apr  6 16:47 wtmp
chsnv@acer:/var/log$
```

ls -lhS

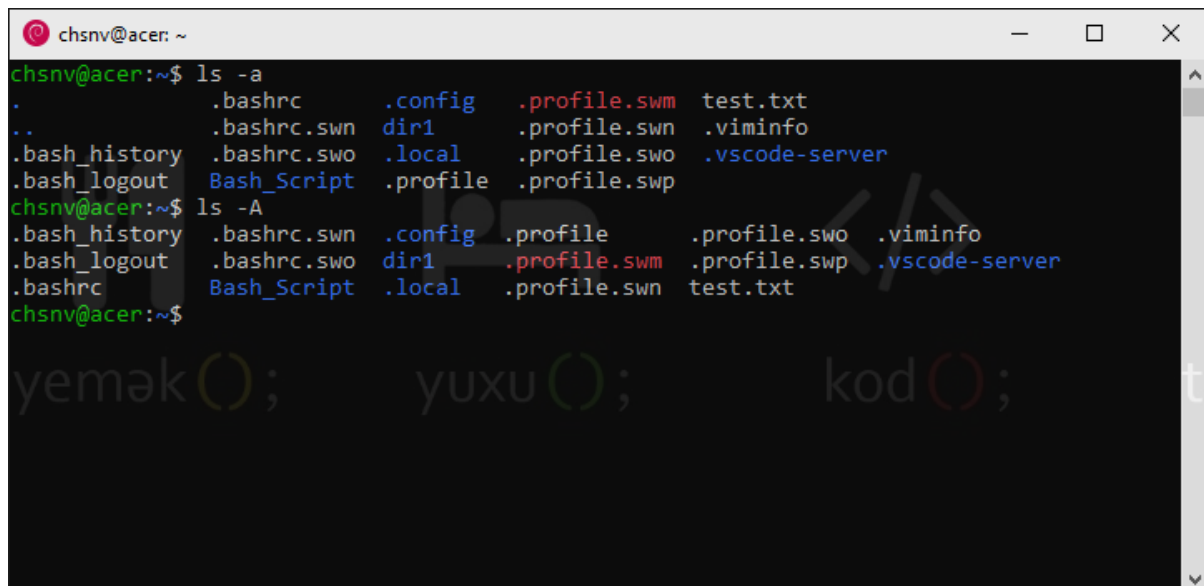
ls -lhS - direktoriyanın insan oxuyacaq formatda detalları ilə və ölçüyə görə siyahısı.



```
chsnv@acer: /var/log$ ls -lhS
total 620K
-rw-rw-r-- 1 root utmp      286K May  1 05:09 lastlog
-rw-r--r-- 1 root root     183K May 13 23:29 dpkg.log
-rw-r--r-- 1 root root      71K Apr  6 16:48 bootstrap.log
-rw-r--r-- 1 root root      32K May  1 05:09 faillog
-rw-r--r-- 1 root docker    23K May  1 06:14 docker.log
-rw-r--r-- 1 root root      17K May  1 05:19 alternatives.log
drwxr-xr-x 1 root root      512 May 13 23:29 apt
drwxr-x--- 1 root adm       512 Jun  8 2019 unattended-upgrades
-rw-rw---- 1 root utmp      384 Apr 20 08:05 bttmp
-rw-rw-r-- 1 root utmp        0 Apr  6 16:47 wttmp
chsnv@acer: /var/log$
```

ls -a və ls -A

ls -a - hazırki direktoriyanın və gizli fayl(ların) siyahısı.



```
chsnv@acer: ~$ ls -a
.      .bashrc      .config      .profile.swm  test.txt
..     .bashrc.swn  dir1         .profile.swn  .viminfo
.bash_history .bashrc.swo  .local       .profile.swo  .vscode-server
.bash_logout Bash_Script  .profile     .profile.swp
chsnv@acer: ~$ ls -A
.bash_history .bashrc.swn .config .profile .profile.swo .viminfo
.bash_logout .bashrc.swo dir1 .profile.swm .profile.swp .vscode-server
.bashrc      Bash_Script .local .profile.swn test.txt
chsnv@acer: ~$
```

Qeyd: fərq **-A** kapital ilə başlasa "." nəzərə alınmır.

cp

cp - fayl(ların) hədəf direktoriyada nüsxə çıxarılma.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script dir1 test.txt  
chsnv@acer:~$ cp test.txt test.txt.nüsxə|
```

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script dir1 test.txt  
chsnv@acer:~$ cp test.txt test.txt.nüsxə  
chsnv@acer:~$ ls  
Bash_Script dir1 test.txt test.txt.nüsxə  
chsnv@acer:~$
```

mv

mv - fayl(ların) hədəf direktoriyaya köçürülmə.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script dir1 test.txt test.txt.nüsxə  
chsnv@acer:~$ mv test.txt.nüsxə dir1/
```

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script dir1 test.txt test.txt.nüsxə  
chsnv@acer:~$ mv test.txt.nüsxə dir1/  
chsnv@acer:~$ ls dir1/  
dir2 test.txt.nüsxə  
chsnv@acer:~$ |
```

Qeyd: İsyifadə qaydası “full path” tam yol yazmadan hazırki qovluq ya fayl adı qeyd edilə bilər. Tam yol aşağıdakı qayda ilə qeyd edildikdə əmr icra edilir.

cp <hədəf yol/fayl adı> <hədəf yol/fayl adı>

mv <hədəf yol/fayl adı> <hədəf yol/fayl adı>

rm -rf

rm -rf - qovluq(ları) ya fay(ları) tamamilə silmə.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script  dir1  file1.txt  test.txt  
chsnv@acer:~$ rm -rf dir1 file1.txt  
chsnv@acer:~$ ls  
Bash_Script  test.txt  
chsnv@acer:~$ |
```

Qeyd: Bu əmrə diqqət yetirin, çünki vacib sənədləridə silə bilər. Buna görə **-rf** “remove force” istifadə etməzdən əvvəl həmişə iki dəfə düşünün. Ya da istəyə bağlı **alias** ilə **-i** kimi əmrdə konfigurasiya edilə bilər.

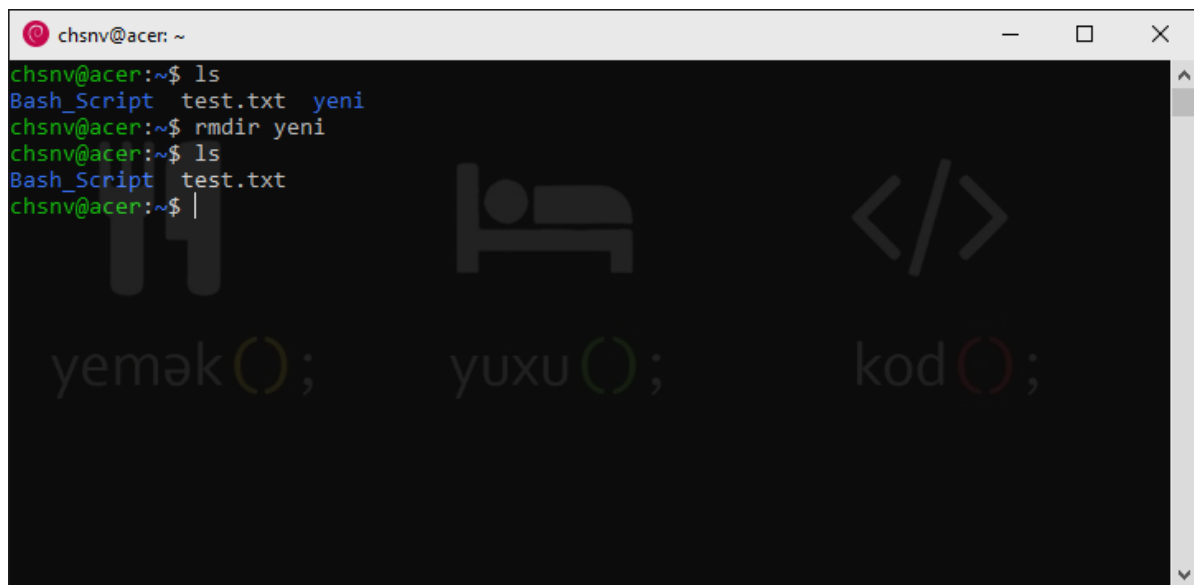
mkdir

mkdir – “make **direktory**” yeni boş qovluq(ları) əlavə edir.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script  test.txt  
chsnv@acer:~$ mkdir yeni  
chsnv@acer:~$ ls  
Bash_Script  test.txt  yeni  
chsnv@acer:~$
```

rmmdir

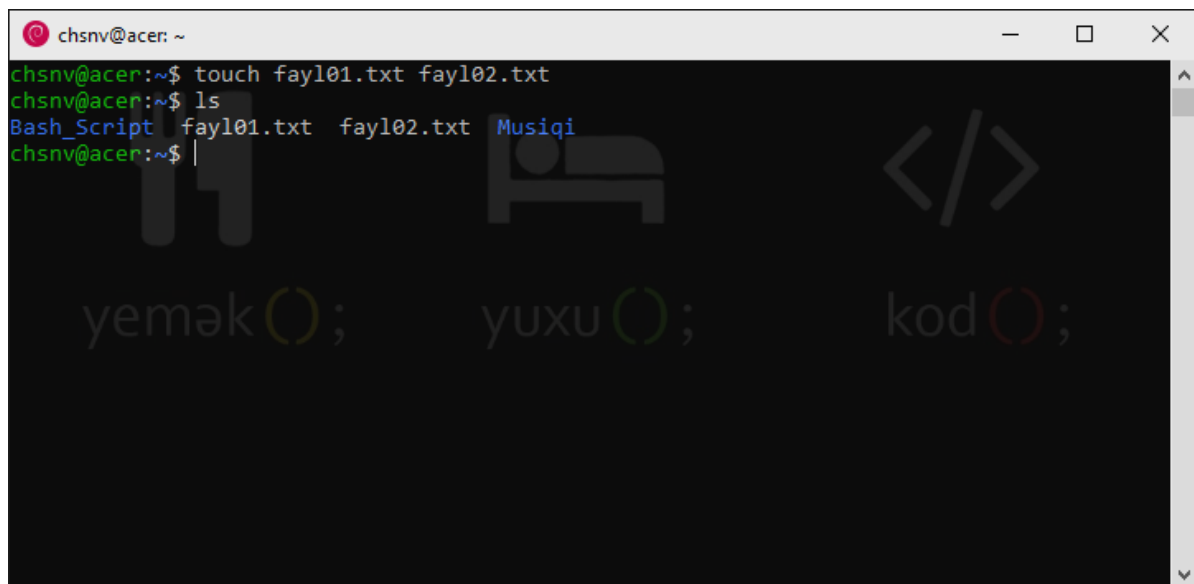
rmmdir – “remove directory” yalnız boş qovluq(ları) silir.



```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script test.txt yeni  
chsnv@acer:~$ rmmdir yeni  
chsnv@acer:~$ ls  
Bash_Script test.txt  
chsnv@acer:~$ |
```

touch

touch - yeni boş fayl(lar) əlavə edir.



```
chsnv@acer: ~  
chsnv@acer:~$ touch fayl01.txt fayl02.txt  
chsnv@acer:~$ ls  
Bash_Script fayl01.txt fayl02.txt Musiqi  
chsnv@acer:~$ |
```

Qeyd: touch ilə fayl(ların) sonluğu istəyə görə seçilə bilər. Burada “.txt” və digər sənəd tipi qeyd edilməsi mümkündür.

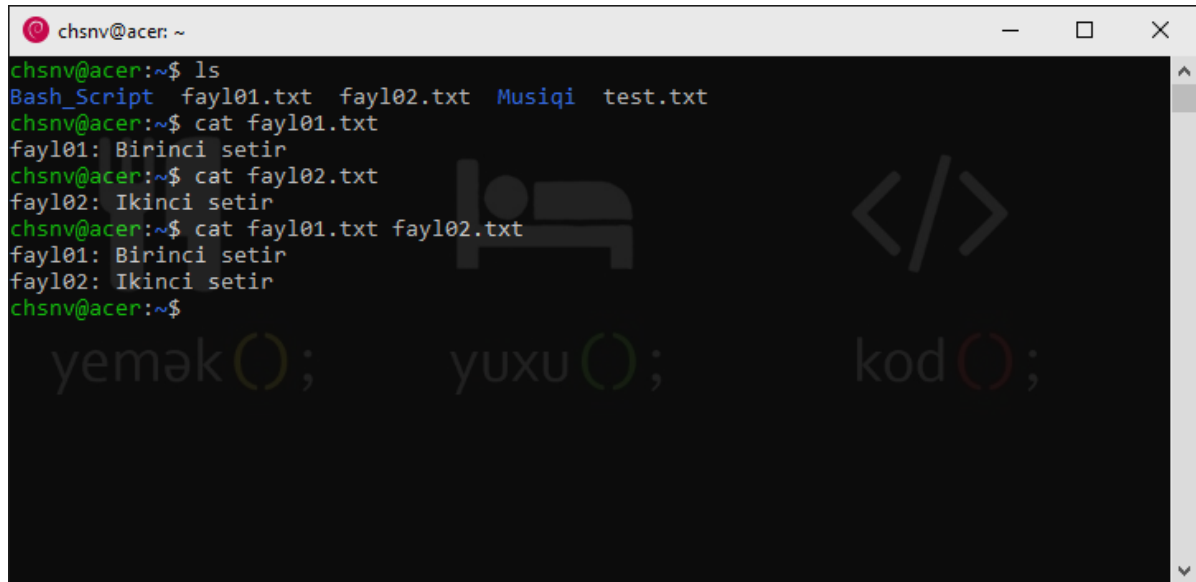
wildcat

wildcat * - ulduz operatoru istifadə qaydaları.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script fayl01.txt fayl02.txt fayl03.txt mahnıA.mp3 mahnıB.mp3 test.txt  
chsnv@acer:~$ rm fayl*.txt  
chsnv@acer:~$ ls  
Bash_Script mahnıA.mp3 mahnıB.mp3 test.txt  
chsnv@acer:~$ |  
  
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script mahnıA.mp3 mahnıB.mp3 Musiqi test.txt  
chsnv@acer:~$ mv *.mp3 ~/Musiqi/cls  
  
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script mahnıA.mp3 mahnıB.mp3 Musiqi test.txt  
chsnv@acer:~$ mv *.mp3 ~/Musiqi/  
chsnv@acer:~$ ls Musiqi/  
mahnıA.mp3 mahnıB.mp3  
chsnv@acer:~$
```

cat

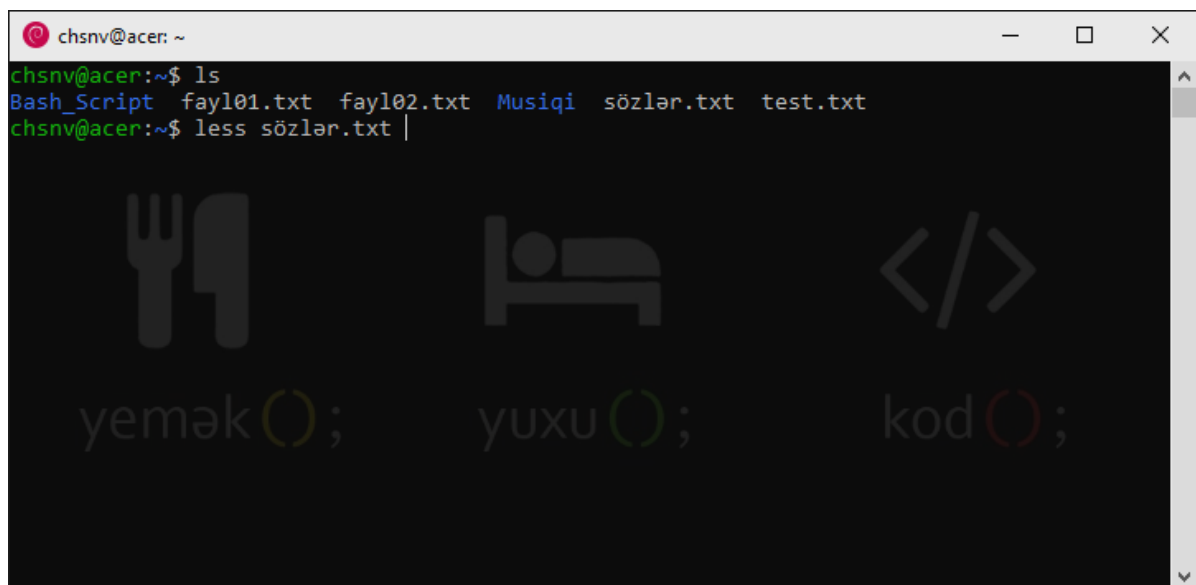
cat - funksiyası faylları birləşdirmək olan və sonra nəticəni ekrana çıxarma olsa da tək bir fayl istifadəsində sadəcə içərisindəki yazıları ekrana çıxarır.

A terminal window titled 'chsnv@acer: ~' showing the execution of the 'cat' command. The user first lists files with 'ls', showing 'Bash_Script', 'fayl01.txt', 'fayl02.txt', 'Musiqi', and 'test.txt'. Then, they use 'cat' to display the contents of 'fayl01.txt' (output: 'Birinci setir'), 'fayl02.txt' (output: 'Ikinci setir'), and finally concatenate 'fayl01.txt' and 'fayl02.txt' (output: 'Birinci setir' followed by 'Ikinci setir'). The terminal has a dark background with a watermark of a person sleeping and the text 'yemək(); yuxu(); kod();'.

```
chsnv@acer:~$ ls
Bash_Script  fayl01.txt  fayl02.txt  Musiqi  test.txt
chsnv@acer:~$ cat fayl01.txt
fayl01: Birinci setir
chsnv@acer:~$ cat fayl02.txt
fayl02: Ikinci setir
chsnv@acer:~$ cat fayl01.txt fayl02.txt
fayl01: Birinci setir
fayl02: Ikinci setir
chsnv@acer:~$
```

less

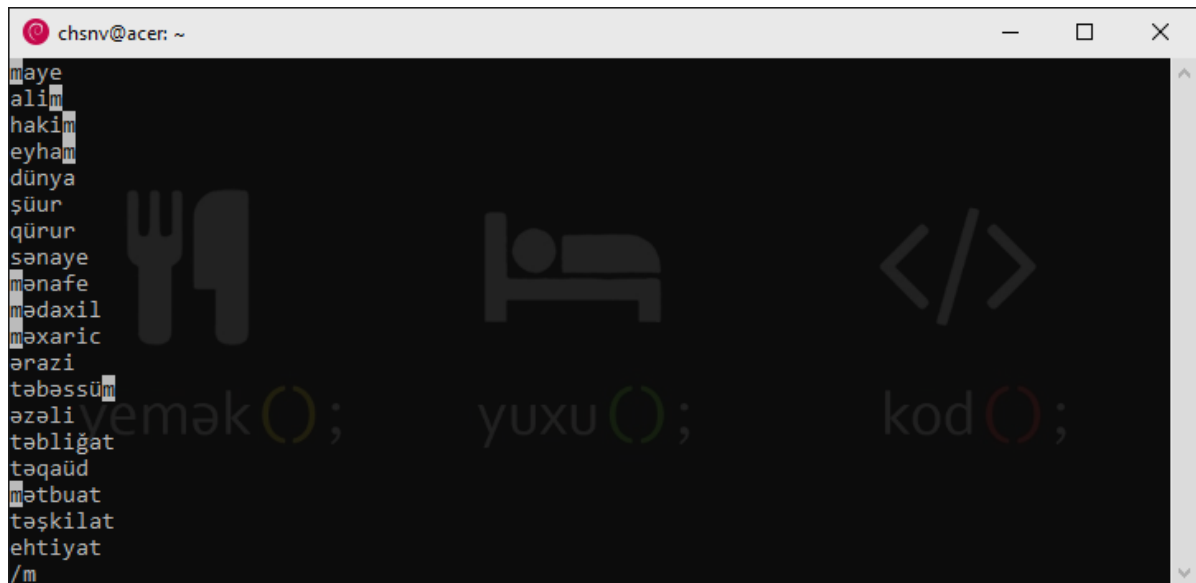
less - mətn daxilində gəzinməyə yararır.

A terminal window titled 'chsnv@acer: ~' showing the execution of the 'less' command. The user lists files with 'ls', showing 'Bash_Script', 'fayl01.txt', 'fayl02.txt', 'Musiqi', 'sözlər.txt', and 'test.txt'. Then, they use 'less' to view the contents of 'sözlər.txt'. The terminal has a dark background with a watermark of a person sleeping and the text 'yemək(); yuxu(); kod();'.

```
chsnv@acer:~$ ls
Bash_Script  fayl01.txt  fayl02.txt  Musiqi  sözlər.txt  test.txt
chsnv@acer:~$ less sözlər.txt |
```

less

"string" tipləri arasında axtarış etməyə imkan verən əmrdir.



grep

grep – fayl(ları), digər əmr(ləri) və mətn(ləri) çıxışı filtirləmək üçün işlədilir.

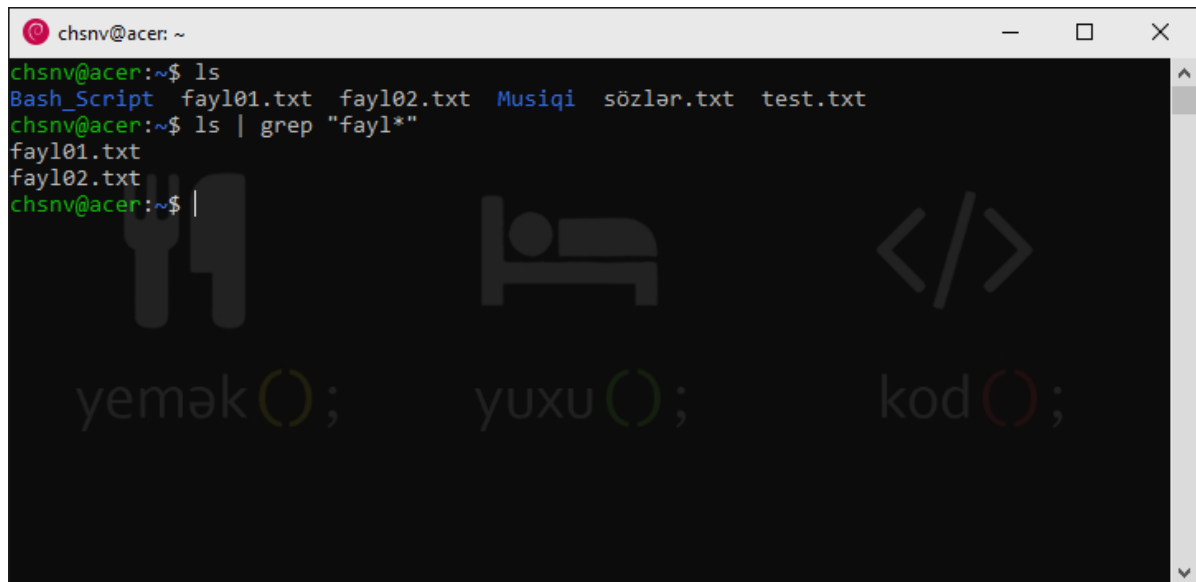


Çıxışı yönləndirmə "Output Redirection"

Normalda bir əmrin çıxışı ekrana yazdırılır, lakin yenidən yönləndirmə operatorları ilə dəyişə bilərik.

"|" çıxışı bir əmrdən digərinə yönləndirmək üçün istifadə olunur. Buna kəməş yəni "piping" də deyilir

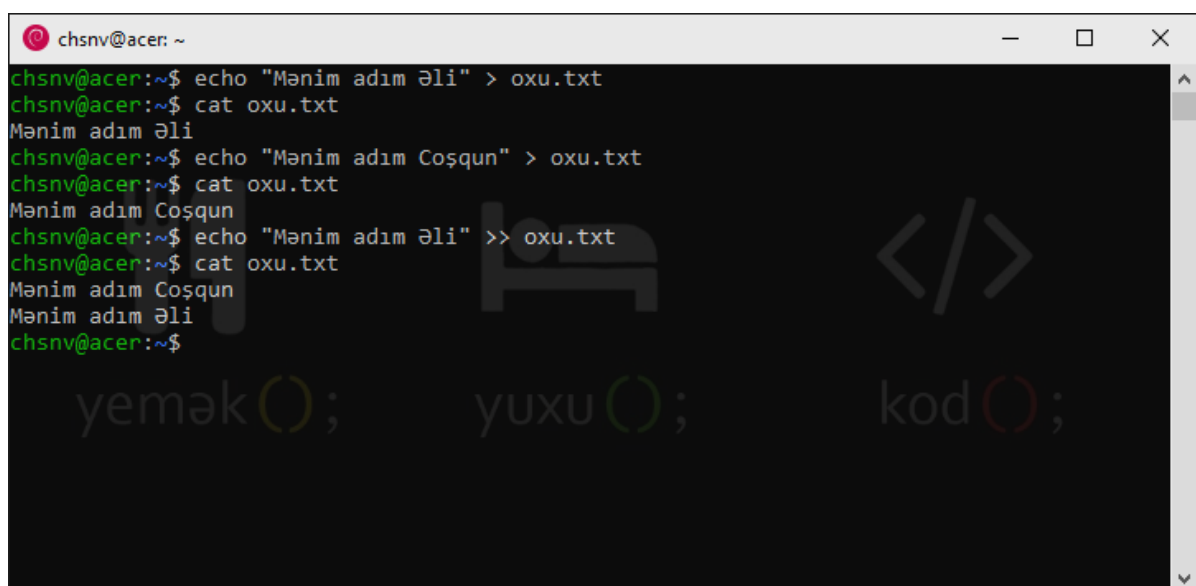
`-ls | grep "fayl*"`



```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script fayl01.txt fayl02.txt Musiqi sözlər.txt test.txt  
chsnv@acer:~$ ls | grep "fayl*"  
fayl01.txt  
fayl02.txt  
chsnv@acer:~$ |
```

`echo "Mətin" > oxu.txt` çıxışı yəni tətbiq edilən bir əmri fayl daxilində üzərinə qeyd etmək üçün istifadə edilir.

`echo "Mətin" >> oxu.txt` çıxışı yəni tətbiq edilən bir əmri fayl daxilində yeni bir sətərə qeyd etmək üçün istifadə edilir.



```
chsnv@acer: ~  
chsnv@acer:~$ echo "Mənim adım Əli" > oxu.txt  
chsnv@acer:~$ cat oxu.txt  
Mənim adım Əli  
chsnv@acer:~$ echo "Mənim adım Coşqun" > oxu.txt  
chsnv@acer:~$ cat oxu.txt  
Mənim adım Coşqun  
chsnv@acer:~$ echo "Mənim adım Əli" >> oxu.txt  
chsnv@acer:~$ cat oxu.txt  
Mənim adım Coşqun  
Mənim adım Əli  
chsnv@acer:~$
```

find

find - müxtəlif meyarlara görə fayl(ları) axtarmaq üçün istifadə olunur.

```
chsnv@acer: ~  
chsnv@acer:~$ ls  
Bash_Script fayl01.txt fayl02.txt Musiqi oxu.txt sözlər.txt test.txt  
chsnv@acer:~$ find /home -name "fa*.txt"  
/home/chsnv/fayl01.txt  
/home/chsnv/fayl02.txt  
chsnv@acer:~$ find /home -name "fa*.txt"
```

du

du - "disk usage" disk istifadəsi direktoriya və bütün qovluqlar daxil olmaqla ölçüləri barədə məlumatı bildirir.

```
chsnv@acer: ~  
chsnv@acer:~$ pwd  
/home/chsnv  
chsnv@acer:~$ ls -lh  
total 4.0K  
drwxr-xr-x 1 chsnv chsnv 512 May 11 14:39 Bash_Script  
-rw-r--r-- 1 chsnv chsnv 23 May 16 16:42 fayl01.txt  
-rw-r--r-- 1 chsnv chsnv 21 May 16 16:43 fayl02.txt  
drwxr-xr-x 1 chsnv chsnv 512 May 16 16:29 Musiqi  
-rw-r--r-- 1 chsnv chsnv 39 May 16 22:19 oxu.txt  
-rw-r--r-- 1 chsnv chsnv 766 May 16 18:46 sözlər.txt  
-rw-r--r-- 1 chsnv chsnv 50 May 16 20:17 test.txt  
chsnv@acer:~$ du -s  
141318 .  
chsnv@acer:~$ du -sh  
139M .  
chsnv@acer:~$
```

vi

vi - "text editor" **nano**, **vim** və digərləri mətn redaktorudur. [vi\(kopya kağızı\)](#)

[illegible]

Qeyd: *gedit* yəni “graphic edit” və digərlərini GUI visual olaraq terminal üzərindən deyil Windows *notepad* text redaktoru kimi düşünə bilərsiniz.

ln

In -s - qısa yolun qurulması əmri */home/chsnv/sözlər.txt* tam qeyd edilməlidir səbəb yer dəyişmə edərkən, olası hər hansı **xətanın** qarşısını almaq.

```
chsnv@acer: ~$ ls
Bash_Script fayl01.txt fayl02.txt Musiqi oxu.txt sözlər.txt test.txt
chsnv@acer:~$ ln -s /home/chsnv/sözlər.txt qisaYol
chsnv@acer:~$ ls -l
total 4
drwxr-xr-x 1 chsnv chsnv 512 May 11 14:39 Bash_Script
-rw-r--r-- 1 chsnv chsnv 23 May 16 16:42 fayl01.txt
-rw-r--r-- 1 chsnv chsnv 21 May 16 16:43 fayl02.txt
drwxr-xr-x 1 chsnv chsnv 512 May 20 03:47 Musiqi
-rw-r--r-- 1 chsnv chsnv 39 May 16 22:19 oxu.txt
lrwxrwxrwx 1 chsnv chsnv 24 May 20 03:49 qisaYol -> /home/chsnv/sözlər.txt
-rw-r--r-- 1 chsnv chsnv 766 May 16 18:46 sözlər.txt
-rw-r--r-- 1 chsnv chsnv 50 May 16 20:17 test.txt
chsnv@acer:~$
```


alias

alias - linux əməllərini şəxsiləşdirməkdir. Bir çox əmr bircə icrası “|” buna “multiple piping” deyilir.

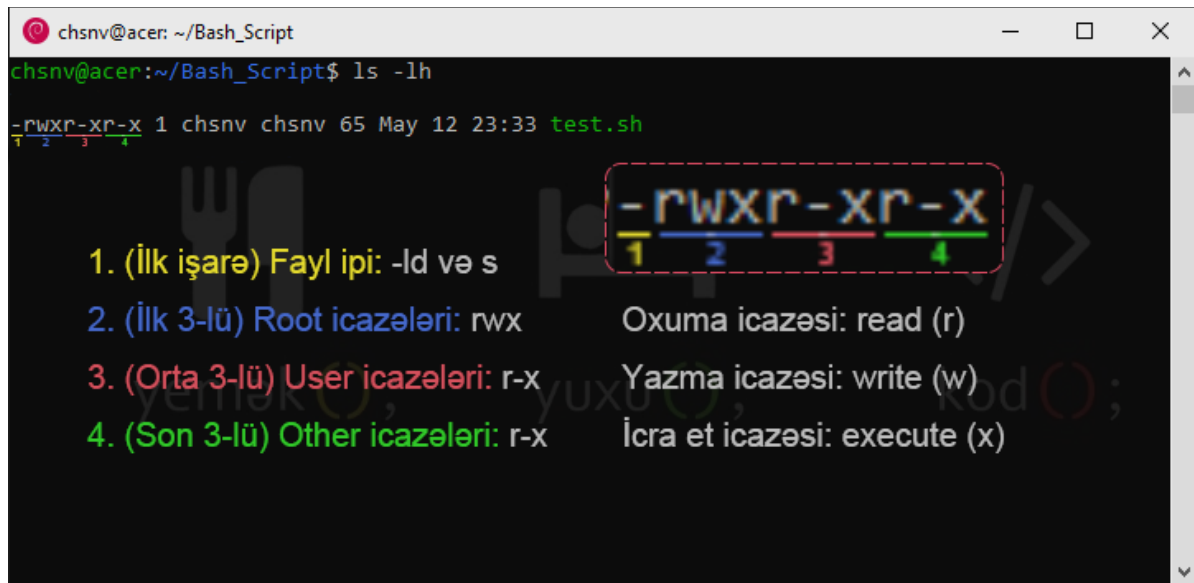
```
chsnv@acer: ~  
chsnv@acer:~$  
chsnv@acer:~$ alias myip='ip a | grep "inet" | head -n 3 | tail -n 1 | cut -d "t" -f 2 |  
cut -d "/" -f 1'  
chsnv@acer:~$  
chsnv@acer:~$ myip  
192.168.1.38  
chsnv@acer:~$
```

sed

sed - əmri ilə çıxışı manipulyasiya etmək və nəticəsini *tarix.txt* kimi saxlamaq.

```
chsnv@acer: ~  
chsnv@acer:~$ date  
Thu 20 May 2021 04:52:27 AM +04  
chsnv@acer:~$ date | sed 's/Thu/Cümə axşamı/' | cut -d "A" -f 1  
Cümə axşamı 20 May 2021 04:52:43  
chsnv@acer:~$ date | sed 's/Thu/Cümə axşamı/' | cut -d "A" -f 1 &> tarix.txt  
chsnv@acer:~$  
chsnv@acer:~$ cat tarix.txt  
Cümə axşamı 20 May 2021 04:52:51  
chsnv@acer:~$
```

“Permissions” icazələr



```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ ls -lh
-rwxr-xr-x 1 chsnv chsnv 65 May 12 23:33 test.sh
```

1. (İlk işarə) Fayl ipi: -ld və s

2. (İlk 3-lü) Root icazələri: rwx

3. (Orta 3-lü) User icazələri: r-x

4. (Son 3-lü) Other icazələri: r-x

Oxuma icazəsi: read (r)

Yazma icazəsi: write (w)

İcra et icazəsi: execute (x)

Qeyd: Nəticəni belə oxuya bilərik *test.sh* faylı *chsnv* adlı istifadəçiyə məxsusdur.

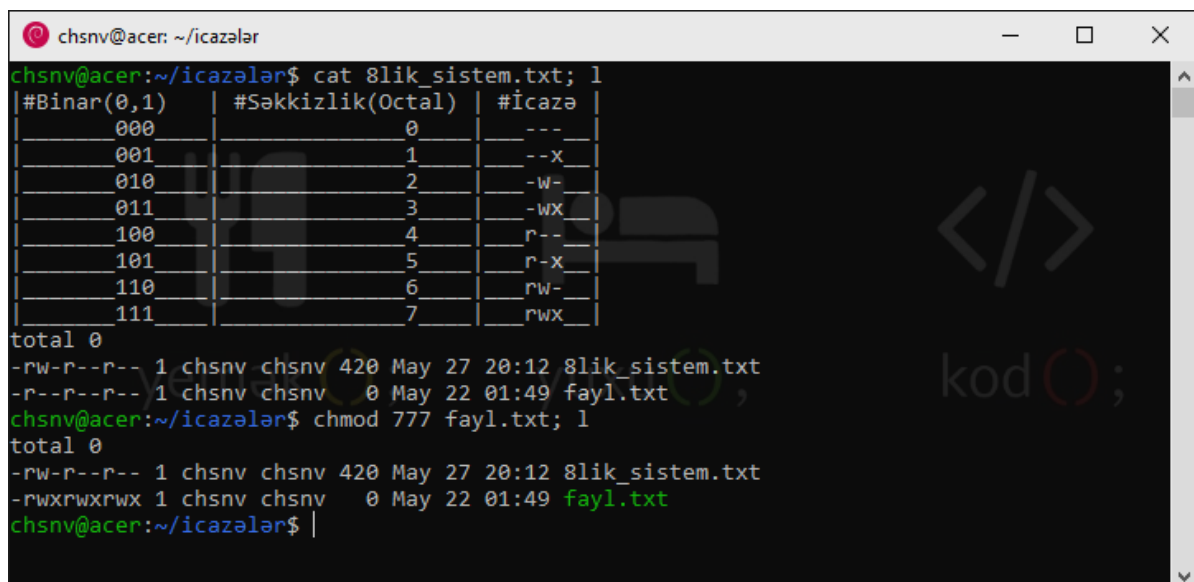
İstifadəçi *chsnv* bu faylı oxumaq, yazmaq və icra etmək hüququna malikdir.

Fayl özü *chsnv* qrupuna məxsusdur.

Yəni ki, *chsnv* qrupunun üzvləri bu faylı oxuya və icra edə bilərlər.

Sonda bu faylı hər kəs oxuya və icra edə bilər.

8-lik sistem



```
chsnv@acer: ~/icazələr
chsnv@acer:~/icazələr$ cat 8lik_sistem.txt; l
#Binar(0,1) | #Səkkizlik(Octal) | #İcazə
|-----|
| 000 | 0 | --- |
| 001 | 1 | --X |
| 010 | 2 | -W- |
| 011 | 3 | -WX |
| 100 | 4 | r-- |
| 101 | 5 | r-X |
| 110 | 6 | rW- |
| 111 | 7 | rWX |
total 0
-rw-r--r-- 1 chsnv chsnv 420 May 27 20:12 8lik_sistem.txt
-r--r--r-- 1 chsnv chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ chmod 777 fayl.txt; l
total 0
-rw-r--r-- 1 chsnv chsnv 420 May 27 20:12 8lik_sistem.txt
-rwxrwxrwx 1 chsnv chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ |
```

Qeyd: *chmod 654 (rw-)(r-x)(r--)* yəni sahibi “owner” sənədi **oxuya yaz**a, grupdakılar **oxuya və icra edə** sonda digərləri “others” ancaq **oxuya** bilərlər.

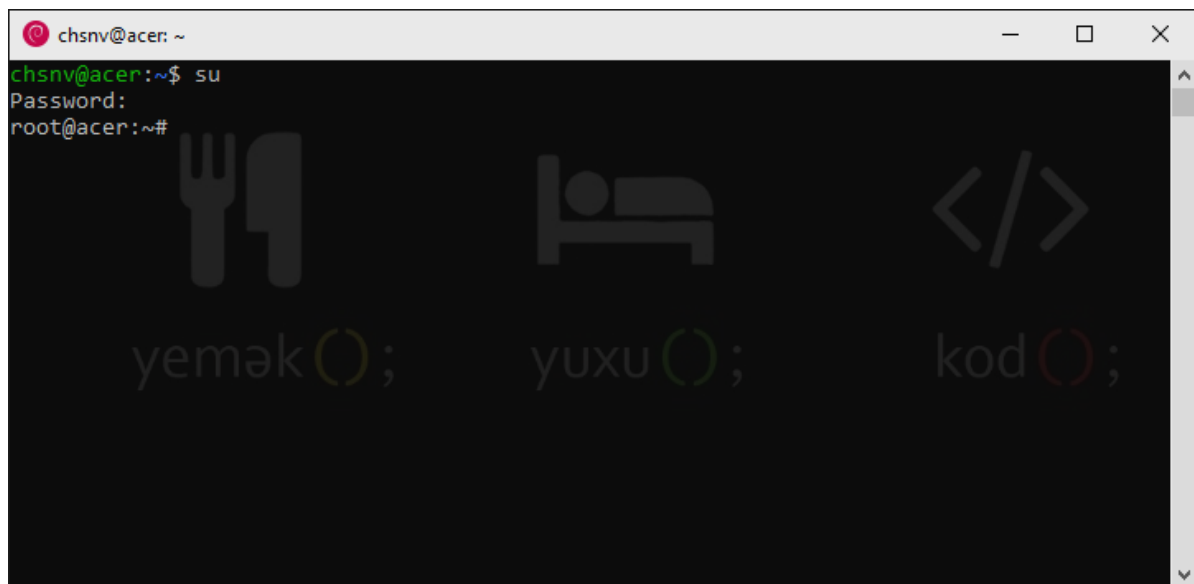
Nədir bu icazələr?

Linux əməliyyat sistemi çox istifadəçi sistemidir. Yəni eyni anda birdən çox istifadəçi kompüterdən istifadə edə bilər.

Bunu asanlaşdırmaq üçün bir istifadəçinin hərəkətlərinin kompüteri çökməsinə və ya bir istifadəçinin digərinin sənədlərinə müdaxilə etməsinə icazə verməmək məqsədli icazələr “permissions” istifadə edilir.

su

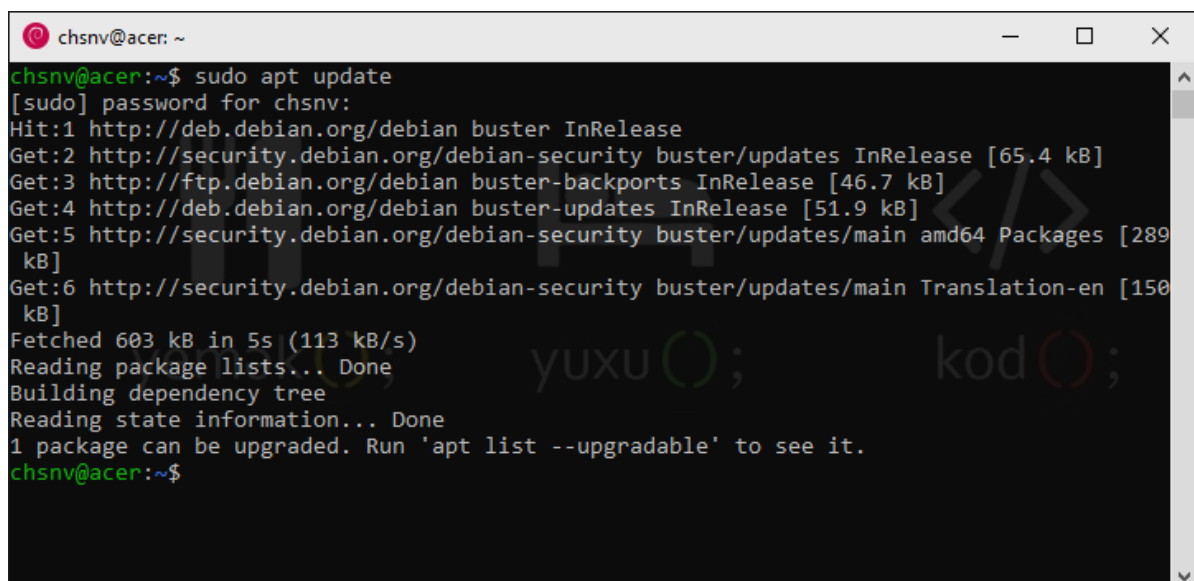
su - super istifadəçiyə keçə bilərsiniz.



```
chsnv@acer: ~$ su
Password:
root@acer: ~#
```

sudo

sudo - super istifadəçi olaraq bir əmr çalışdırı bilərsiniz.



```
chsnv@acer: ~$ sudo apt update
[sudo] password for chsnv:
Hit:1 http://deb.debian.org/debian buster InRelease
Get:2 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]
Get:3 http://ftp.debian.org/debian buster-backports InRelease [46.7 kB]
Get:4 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]
Get:5 http://security.debian.org/debian-security buster/updates/main amd64 Packages [289 kB]
Get:6 http://security.debian.org/debian-security buster/updates/main Translation-en [150 kB]
Fetched 603 kB in 5s (113 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
chsnv@acer: ~$
```

chmod

chmod - *chmod* <u/g/o/a> <+/-> <r/w/x> fayl.txt giriş hüquqlarını dəyişdirə bilərsiniz.

```
chsnv@acer: ~/icazələr
chsnv@acer:~$ mkdir icazələr && cd icazələr && touch fayl.txt && ls -l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ chmod u-w fayl.txt
chsnv@acer:~/icazələr$ ls -l
total 0
-r--r--r-- 1 chsnv chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$
```

Qeyd: Sənədin icazələrini görmək üçün *ls -l* istifadə etmək kifayətdir.

u "User" istifadəçi

g "Group" qrup

o "Other" başqaları

a "All" hər kəs üçün deməkdir.

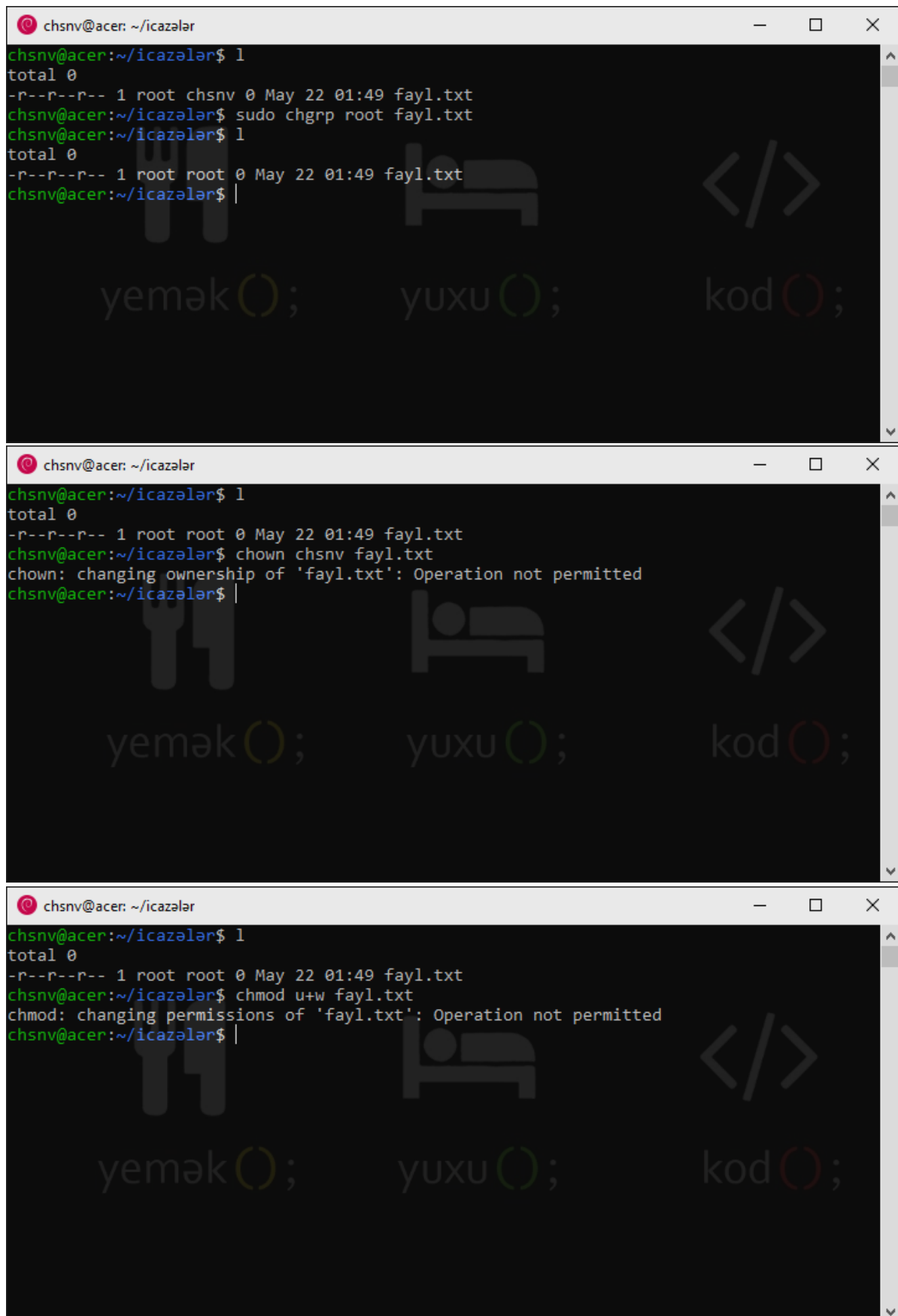
chown

chown – fayl(ların) sahibliyini dəyişdirər.

```
chsnv@acer: ~/icazələr
chsnv@acer:~/icazələr$ l
total 0
-r--r--r-- 1 chsnv chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ sudo chown root fayl.txt
[sudo] password for chsnv:
chsnv@acer:~/icazələr$ l
total 0
-r--r--r-- 1 root chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ |
```

chgrp

chgrp – fayl(ların) qrupunu dəyişdirər.



The image displays three terminal windows from a user named 'chsnv' on a system named 'acer', located in the directory '~/icazələr'. Each window shows a sequence of commands and their outputs.

Terminal 1:

```
chsnv@acer: ~/icazələr
chsnv@acer:~/icazələr$ ls
total 0
-r--r--r-- 1 root chsnv 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ sudo chgrp root fayl.txt
chsnv@acer:~/icazələr$ ls
total 0
-r--r--r-- 1 root root 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$
```

Terminal 2:

```
chsnv@acer: ~/icazələr
chsnv@acer:~/icazələr$ ls
total 0
-r--r--r-- 1 root root 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ chown chsnv fayl.txt
chown: changing ownership of 'fayl.txt': Operation not permitted
chsnv@acer:~/icazələr$
```

Terminal 3:

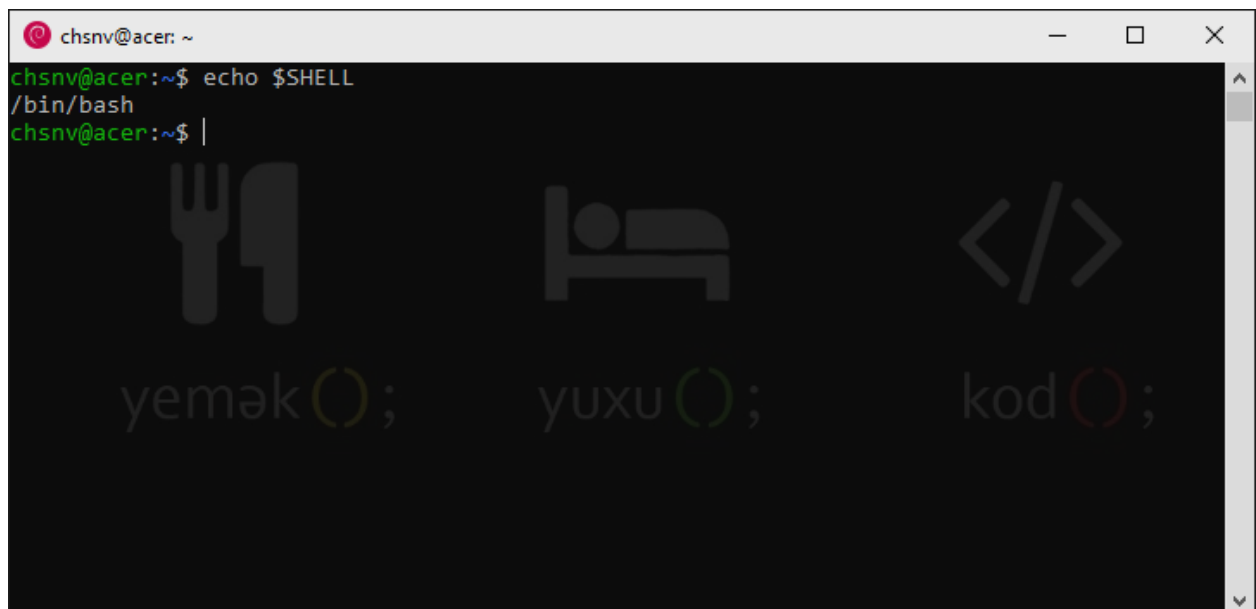
```
chsnv@acer: ~/icazələr
chsnv@acer:~/icazələr$ ls
total 0
-r--r--r-- 1 root root 0 May 22 01:49 fayl.txt
chsnv@acer:~/icazələr$ chmod u+w fayl.txt
chmod: changing permissions of 'fayl.txt': Operation not permitted
chsnv@acer:~/icazələr$
```

Shell Script

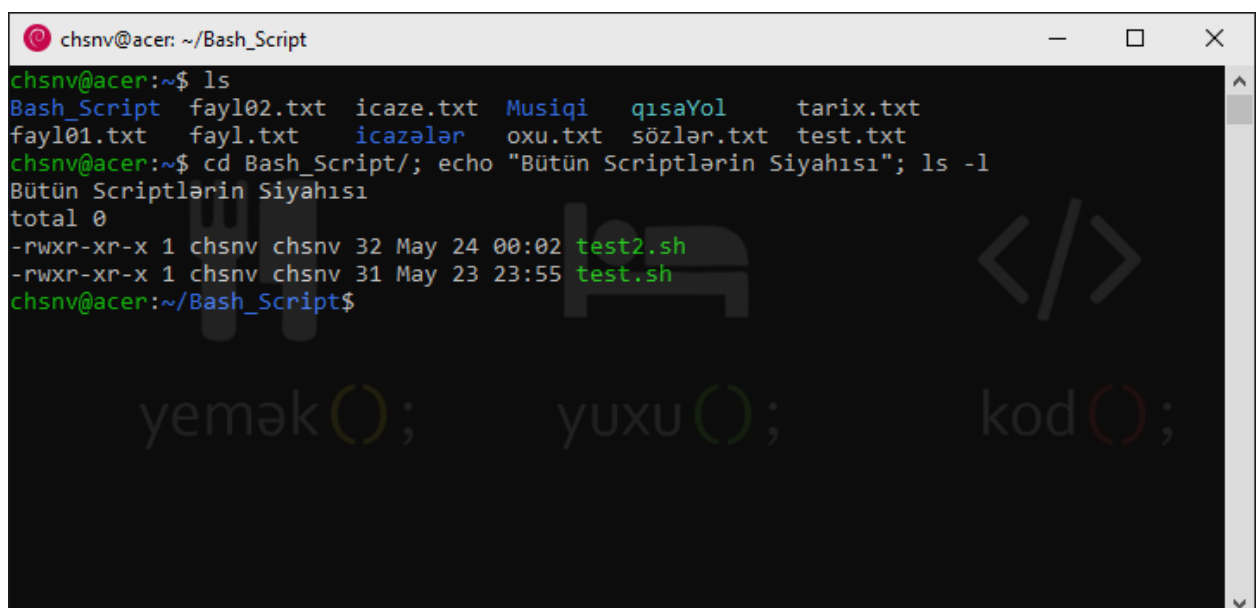
Bash Script'ə giriş

Nədir bu bash script?

"Bourne Again SHell" bash script bir icra edilən "executable" fayl(lar) ya sənəd(lər) sayılır. Bash'in proqramlaşdırma quruluşu var və bash dili istifadə edilir. Bu proqramlama dili ən yaxşısı deyildir təbii ki, əksinə mürəkkəb işlər görmək üçün Python və digər əsas proqramlama dilləri istifadə edilir. Bash bir çox hallarda istifadəsi lazımlı və rahatdır.



```
chsnv@acer: ~  
chsnv@acer:~$ echo $SHELL  
/bin/bash  
chsnv@acer:~$ |
```




```
chsnv@acer: ~/Bash_Script  
chsnv@acer:~$ ls  
Bash_Script fayl02.txt icaze.txt Musiqi qısaYol tarix.txt  
fayl01.txt fayl.txt icazələr oxu.txt sözlər.txt test.txt  
chsnv@acer:~$ cd Bash_Script/; echo "Bütün Scriptlərin Siyahısı"; ls -l  
Bütün Scriptlərin Siyahısı  
total 0  
-rwxr-xr-x 1 chsnv chsnv 32 May 24 00:02 test2.sh  
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh  
chsnv@acer:~/Bash_Script$
```

She-bang

Bash shell istifadəsi zamanı istənilən **vi**, **vim** text redaktoru istifadəsi zamanı öndə **#!** yəni she-bang işarəsi daha sonra mütləq yolu "absolute path" qeyd edilir: **#!/bin/bash**

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ 1
total 0
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ |
```



```
chsnv@acer: ~/Bash_Script
#!/bin/bash

function yemak {
    echo 'Bash Shell'
}

yemak()

"test.sh" 3L, 31C
```

Qeyd: Bash proqramlama ilə əməllər istifadəçi tərəfindən eyni anda icra edilir. Yəni hər dəfə “|”, “&” və “;” istifadəsi gərəkməz bu bir vaxtan sonra sıxıcı hal almağa başlayar. Scripting ilə bunun qarşısını səmərəli şəkildə almaq mümkündür.

/etc/shells

Linux'da hazır yülü və ya sonradan qura birləcəyiniz qabıqlar mövcüddür. Bash isə bunlardan ən çox yayılan və istifadə ediləndir.

```
chsnv@acer: ~  
chsnv@acer:~$ cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
/bin/bash  
/bin/rbash  
/bin/dash  
/bin/csh  
/bin/tcsh  
/usr/bin/tcsh  
/bin/ksh93  
/bin/rksh93  
chsnv@acer:~$ |
```

```
chsnv@acer: ~  
chsnv@acer:~$ echo $0  
-bash  
chsnv@acer:~$ sh  
$ echo $0  
sh  
$ rbash  
chsnv@acer:~$ echo $0  
rbash  
chsnv@acer:~$ dash  
$ echo $0  
dash  
$ tcsh  
acer:~> echo $0  
tcsh  
acer:~> ksh  
$ echo $0  
ksh  
$ exit
```

Qeyd: Linux mövcud olan qabıq yəni shell olaraq Korn, C, TC, Debian Almquist və bir çoxlarını sadalaya bilərik. Aralarından çoxu IBM, BSD linux və digər şirkət, universitetlərin yer adığı qurumlar tərəfindən inkişaf etdirilmişdir.

Şərt Testi "Condition Test"

-eq

n1 -eq n2 "equal" ədəd tipi n1 = n2 bərabərdir.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 5 -eq 5 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ |
```

-ge

n1 -ge n2 "greater or equal" ədəd tipi n1 >= n2 böyük və ya bərabərdir.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 5 -ge 6 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$ [ 5 -ge 5 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$
```

-gt

n1 -gt n2 "greater than" ədəd tipi $n1 > n2$ böyükdür.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 5 -gt 6 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$  
chsnv@acer:~$ [ 6 -gt 5 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ |
```

-le

n1 -le n2 "less than or equal" ədəd tipi $n1 \leq n2$ kiçik və ya bərabərdir.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 5 -le 6 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ [ 5 -le 5 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ [ 6 -le 5 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$ |
```

-lt

n1 -lt n2 "less than" ədəd tipi $n1 < n2$ kiçikdir.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 6 -lt 5 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$ [ 5 -lt 6 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ [ 5 -lt 5 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$ |
```

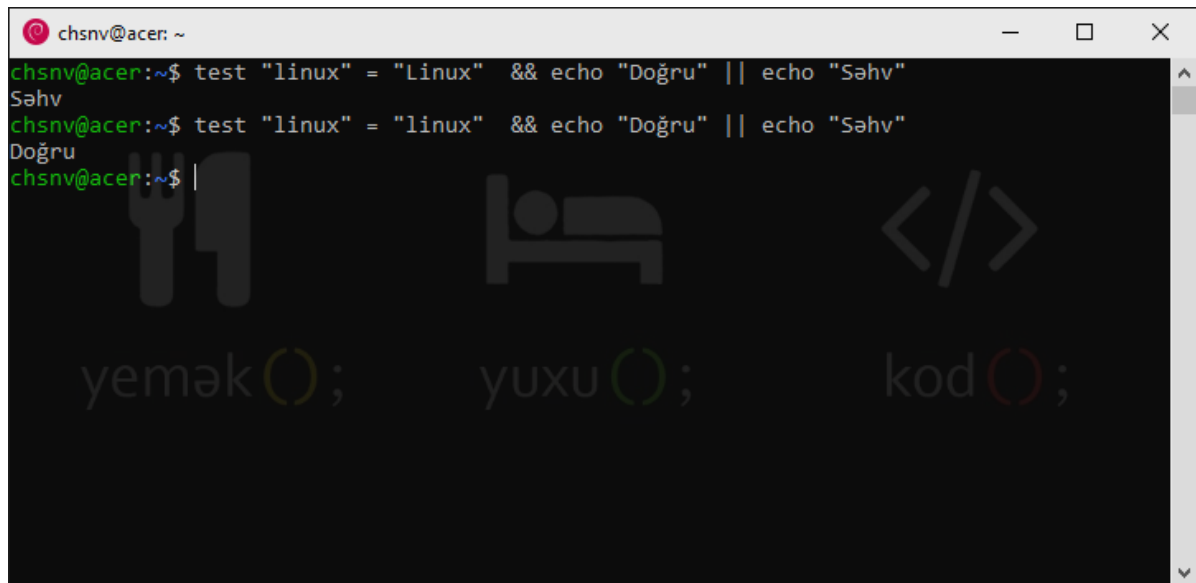
-ne

n1 -ne n2 "not equal" ədəd tipi $n1 \neq n2$ bərabər deyil.

```
chsnv@acer: ~  
chsnv@acer:~$ [ 5 -ne 6 ] && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$ [ 6 -ne 6 ] && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$
```

=

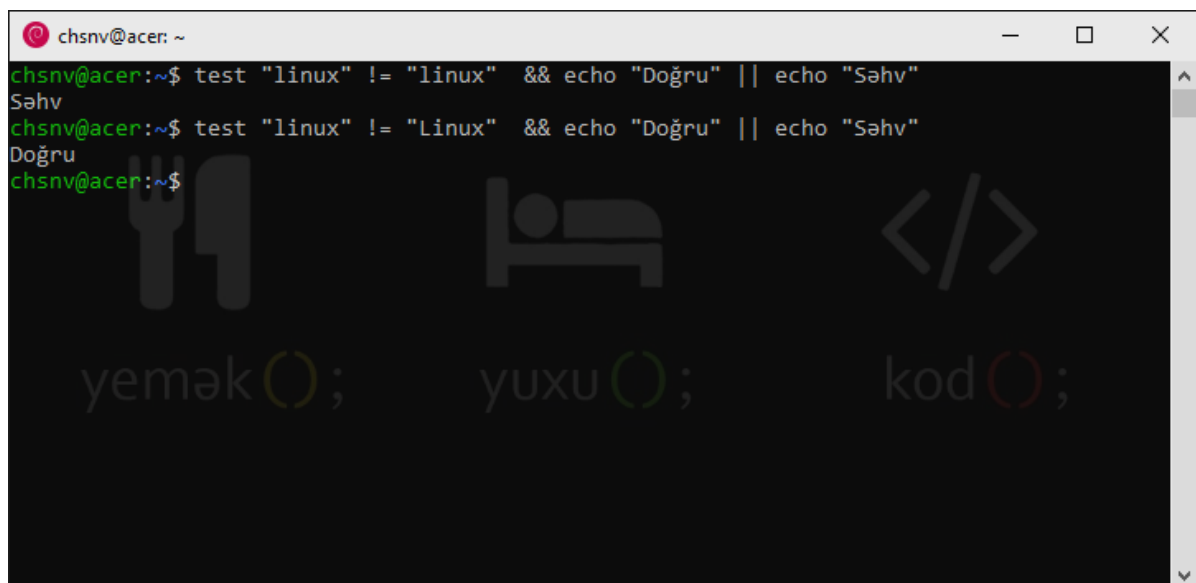
str1 = str2 yazı tipi bərabərliyi yoxlayır.

A terminal window titled 'chsnv@acer: ~' with standard window controls. It shows two test commands. The first command is 'test "linux" = "Linux" && echo "Doğru" || echo "Səhv"', which outputs 'Səhv'. The second command is 'test "linux" = "linux" && echo "Doğru" || echo "Səhv"', which outputs 'Doğru'. The prompt 'chsnv@acer:~\$' is visible at the end of the second line.

```
chsnv@acer:~$ test "linux" = "Linux" && echo "Doğru" || echo "Səhv"
Səhv
chsnv@acer:~$ test "linux" = "linux" && echo "Doğru" || echo "Səhv"
Doğru
chsnv@acer:~$
```

!=

str1 != str2 yazı tipi bərabər olmadığını yoxlayır.

A terminal window titled 'chsnv@acer: ~' with standard window controls. It shows two test commands. The first command is 'test "linux" != "linux" && echo "Doğru" || echo "Səhv"', which outputs 'Səhv'. The second command is 'test "linux" != "Linux" && echo "Doğru" || echo "Səhv"', which outputs 'Doğru'. The prompt 'chsnv@acer:~\$' is visible at the end of the second line.

```
chsnv@acer:~$ test "linux" != "linux" && echo "Doğru" || echo "Səhv"
Səhv
chsnv@acer:~$ test "linux" != "Linux" && echo "Doğru" || echo "Səhv"
Doğru
chsnv@acer:~$
```

<

str1 < str2 yazı tipi kiçik olduğunu yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ [[ "Bir" < "OnBir" ]] && echo "Doğru" || echo "Yalnis"
Doğru
chsnv@acer:~/Bash_Script$ [[ "YüzOnBir" < "OnBir" ]] && echo "Doğru" || echo "Yalnis"
Yalnis
chsnv@acer:~/Bash_Script$
```

>

str1 > str2 yazı tipi böyük olduğunu yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ [[ "YüzOnBir" > "OnBir" ]] && echo "Doğru" || echo "Yalnis"
Doğru
chsnv@acer:~/Bash_Script$ [[ "Bir" > "OnBir" ]] && echo "Doğru" || echo "Yalnis"
Yalnis
chsnv@acer:~/Bash_Script$ |
```

-n

n xarakter uzunluğunun 0 dan böyük olduğunu yoxlayır.

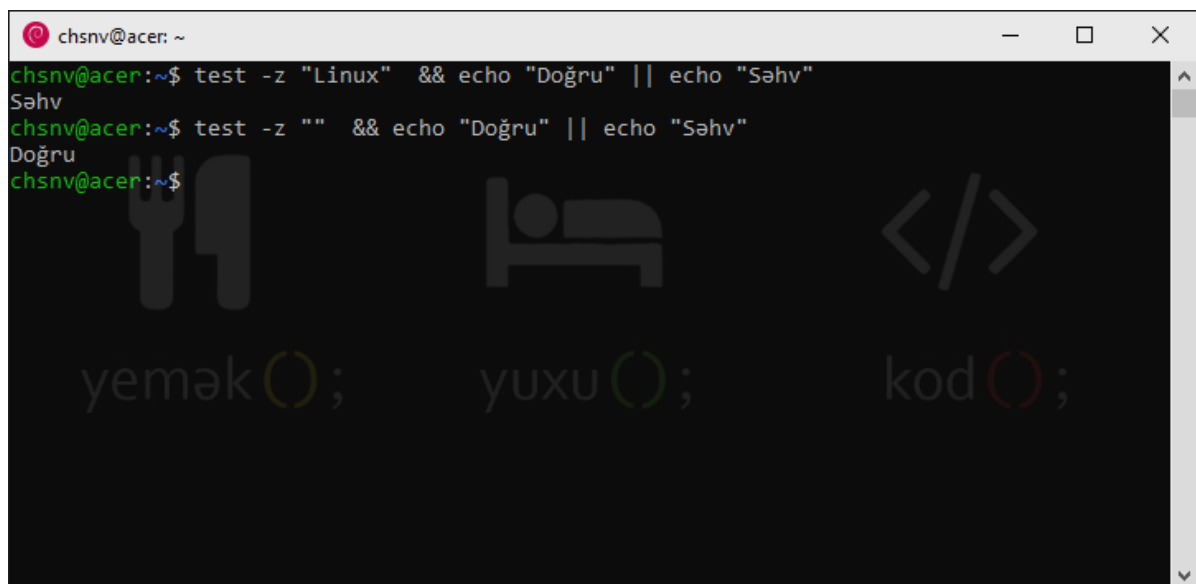


```
chsnv@acer: ~  
chsnv@acer:~$ test -n "Linux" && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$
```

The terminal window shows the command `test -n "Linux" && echo "Doğru" || echo "Səhv"` being executed. The output is `Doğru`, indicating that the string "Linux" is non-empty. The background features a dark theme with icons for a fork and knife (yemək), a bed (yuxu), and code symbols (kod).

-z

z string xarakter uzunluğunun 0 olduğunu yoxlayır.



```
chsnv@acer: ~  
chsnv@acer:~$ test -z "Linux" && echo "Doğru" || echo "Səhv"  
Səhv  
chsnv@acer:~$ test -z "" && echo "Doğru" || echo "Səhv"  
Doğru  
chsnv@acer:~$
```

The terminal window shows two commands. The first is `test -z "Linux" && echo "Doğru" || echo "Səhv"`, which outputs `Səhv` because "Linux" is not an empty string. The second is `test -z "" && echo "Doğru" || echo "Səhv"`, which outputs `Doğru` because the empty string is indeed empty. The background features a dark theme with icons for a fork and knife (yemək), a bed (yuxu), and code symbols (kod).

-d

d file mövcud və direktoriya olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -d qovluq && echo "Doğru" || echo "Səhv"
Doğru
chsnv@acer:~/Bash_Script$ test -d test.sh && echo "Doğru" || echo "Səhv"
Səhv
chsnv@acer:~/Bash_Script$
```

-e

-e fayl və qovluğun mövcud olub ya olmadığını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -e test.sh && echo "Doğru" || echo "Səhv"
Doğru
chsnv@acer:~/Bash_Script$ test -e test.php && echo "Doğru" || echo "Səhv"
Səhv
chsnv@acer:~/Bash_Script$ test -e qovluq && echo "Doğru" || echo "Səhv"
Doğru
chsnv@acer:~/Bash_Script$ test -e fayl && echo "Doğru" || echo "Səhv"
Səhv
chsnv@acer:~/Bash_Script$
```

-f

-f file mövcud olan fayl və ya qovluq tipi olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -f test.sh && echo "Fayl" || echo "Qovluq"
Fayl
chsnv@acer:~/Bash_Script$ test -f qovluq && echo "Fayl" || echo "Qovluq"
Qovluq
chsnv@acer:~/Bash_Script$
```

-r

-r mövcud və oxuma icazəsinin olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
-rwxr-xr-x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -r test.sh && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ chmod -r test.sh
chsnv@acer:~/Bash_Script$ test -r test.sh && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$ |
```


-W

-w mövcud və **yazma** icazəsinin olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
--wx--x--x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -w test.sh && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ chmod -w test.sh
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
---x--x--x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -w test.sh && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$ |
```

-X

-x mövcud və **icra et** icazəsinin olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
---x--x--x 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -x test.sh && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ chmod -x test.sh
chsnv@acer:~/Bash_Script$ l
total 0
drwxr-xr-x 1 chsnv chsnv 512 May 27 05:44 qovluq
----- 1 chsnv chsnv 31 May 23 23:55 test.sh
chsnv@acer:~/Bash_Script$ test -x test.sh && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$
```

-O

-O mövcud sənədin cari istifadəçiyə məxsusluğunu yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 root  root  0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$ test -O köhnə.txt && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ sudo chown root köhnə.txt
chsnv@acer:~/Bash_Script$ test -O köhnə.txt && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 root chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 root root  0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$
```

-G

-G mövcud və cari istifadəçi ilə qurupunun eyni olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$ test -G yeni.txt && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ sudo chgrp root yeni.txt
[sudo] password for chsnv:
chsnv@acer:~/Bash_Script$ test -G yeni.txt && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 chsnv root  0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$
```

-nt

fayl1 **-nt** fayl2 sənədin **yeni** olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$ test yeni.txt -nt köhnə.txt && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ test köhnə.txt -nt yeni.txt && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$
```

-ot

fayl1 **-ot** fayl2 **köhnə** olmasını yoxlayır.

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ l
total 0
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:22 köhnə.txt
-rw-r--r-- 1 chsnv chsnv 0 May 27 07:24 yeni.txt
chsnv@acer:~/Bash_Script$ test yeni.txt -ot köhnə.txt && echo "Doğru" || echo "Yalnış"
Yalnış
chsnv@acer:~/Bash_Script$ test köhnə.txt -ot yeni.txt && echo "Doğru" || echo "Yalnış"
Doğru
chsnv@acer:~/Bash_Script$ |
```

Bash İlk Proqram

vi text redaktoru ilə ilk "Salam Dünya!" və sadə hesab proqramları kodlama

```
chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ cat > SalamDünya.sh <<stp
> #!/bin/bash
> echo "Salam Dünya!"
> stp
chsnv@acer:~/Bash_Script$ chmod 744 SalamDünya.sh
chsnv@acer:~/Bash_Script$ l
total 0
-rwxr--r-- 1 chsnv chsnv 33 May 28 07:53 SalamDünya.sh
chsnv@acer:~/Bash_Script$ ./SalamDünya.sh
Salam Dünya!
chsnv@acer:~/Bash_Script$

chsnv@acer: ~/Bash_Script
#!/bin/bash

n1=2
n2=7

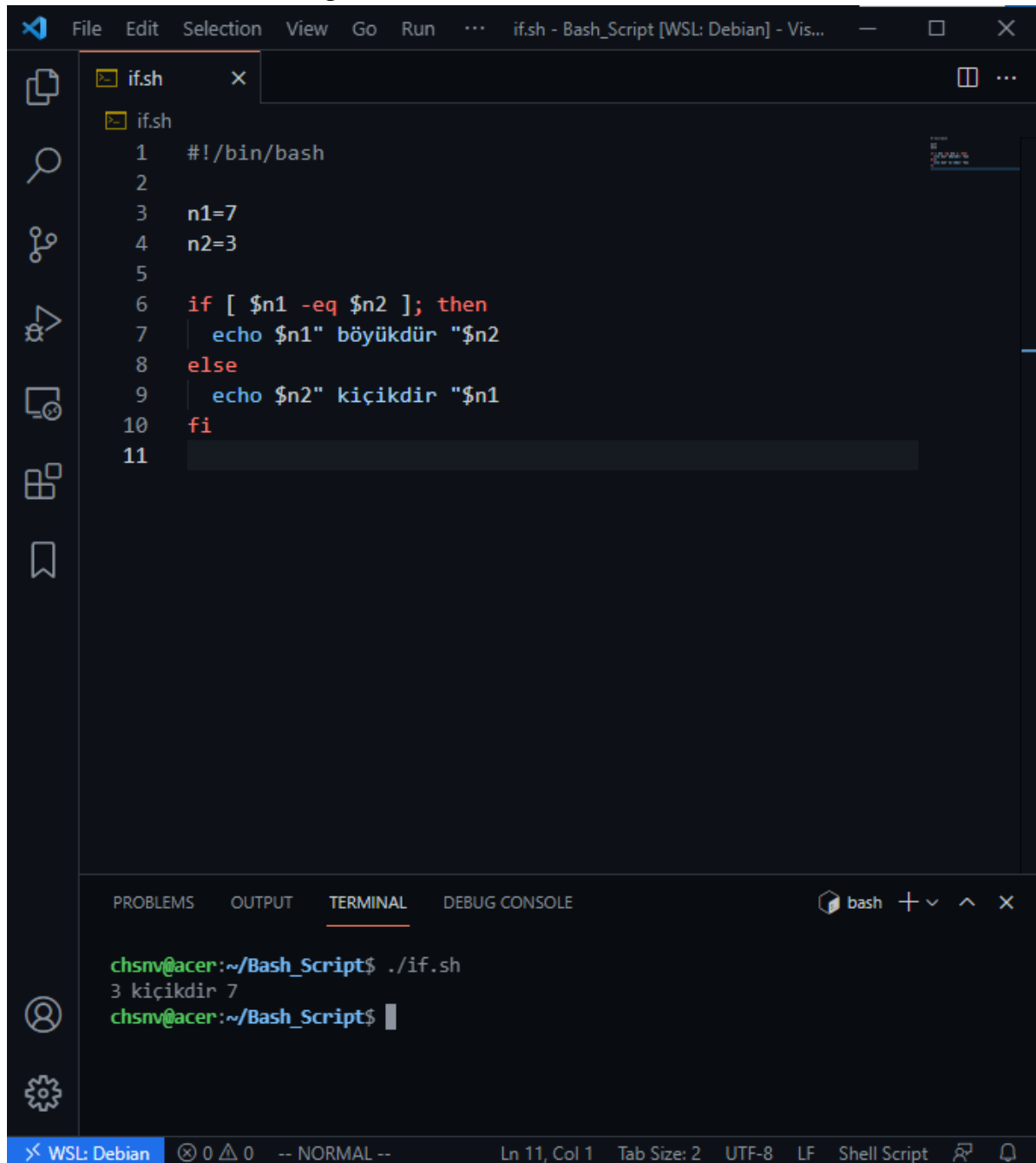
cem=$((n1+n2))
ferq=$((n2-n1))

echo $n1 + "$n2" = "$cem
echo $n2 - "$n1" = "$ferq
~
~
~
~
~
~
~
~
~
~
:wq

chsnv@acer: ~/Bash_Script
chsnv@acer:~/Bash_Script$ vi hesab.sh
chsnv@acer:~/Bash_Script$ l
total 0
-rwxr--r-- 1 chsnv chsnv 109 May 28 08:18 hesab.sh
-rwxr--r-- 1 chsnv chsnv 33 May 28 07:53 SalamDünya.sh
chsnv@acer:~/Bash_Script$
chsnv@acer:~/Bash_Script$
chsnv@acer:~/Bash_Script$
chsnv@acer:~/Bash_Script$ ./hesab.sh
2 + 7 = 9
7 - 2 = 5
chsnv@acer:~/Bash_Script$
```

if Deyimi

Bash sadə **if** strukturu aşağıdakı kimidir:



```
File Edit Selection View Go Run ... if.sh - Bash_Script [WSL: Debian] - Vis...
if.sh
1  #!/bin/bash
2
3  n1=7
4  n2=3
5
6  if [ $n1 -eq $n2 ]; then
7      echo $n1 " böyükdür "$n2
8  else
9      echo $n2 " kiçikdir "$n1
10 fi
11

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
chsnv@acer:~/Bash_Script$ ./if.sh
3 kiçikdir 7
chsnv@acer:~/Bash_Script$
```

Qeyd: **if** **TEST-ƏMRİ**

then

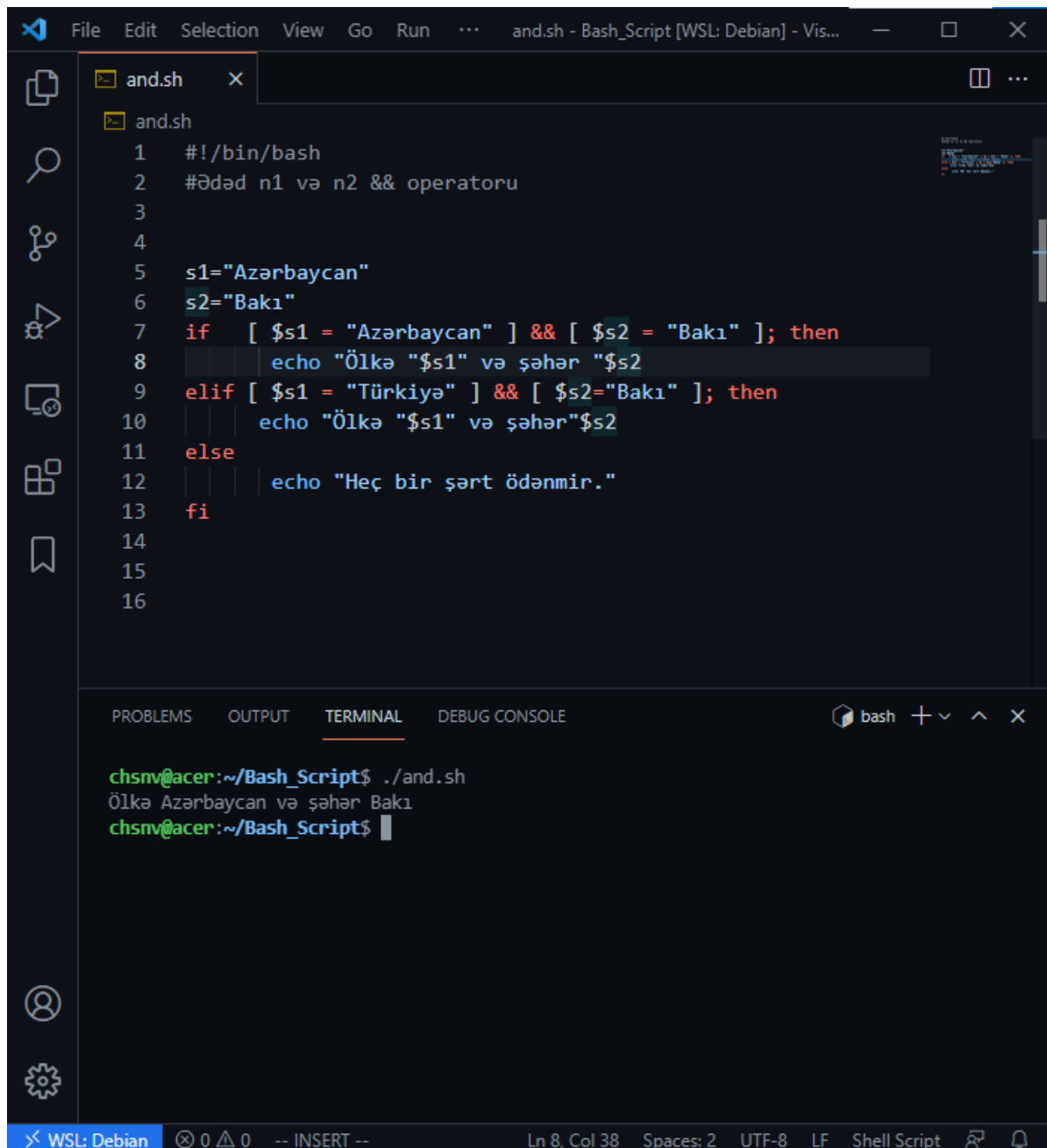
BƏYANATLAR

fi

Əgər **TEST-ƏMRİ** *true* dəyəri qaytarsa , **BƏYANATLAR** icra olacaq. Yox əgər **TEST-ƏMRİ** *false* qaytararsa, heçnə baş verməz. **BƏYANATLAR** nəzərə alınmaz. **if** ifadəsi, **if** açar sözü ilə başlayar, ardınca şərti ifadələr və **then** açar sözü gələr. Sonda bütün bunar **fi** açar sözü ilə bağlanar.

if..else Deyimi

Bash if..else strukturu aşağıdakı kimidir:



```
1  #!/bin/bash
2  #Ədəd n1 və n2 && operatoru
3
4
5  s1="Azərbaycan"
6  s2="Bakı"
7  if [ $s1 = "Azərbaycan" ] && [ $s2 = "Bakı" ]; then
8      echo "Ölkə "$s1" və şəhər "$s2
9  elif [ $s1 = "Türkiyə" ] && [ $s2="Bakı" ]; then
10     echo "Ölkə "$s1" və şəhər"$s2
11 else
12     echo "Heç bir şərt ödənmir."
13 fi
14
15
16
```

```
chsnv@acer:~/Bash_Script$ ./and.sh
Ölkə Azərbaycan və şəhər Bakı
chsnv@acer:~/Bash_Script$
```

Qeyd: if TEST-ƏMRİ

then

BƏYANAT1

else

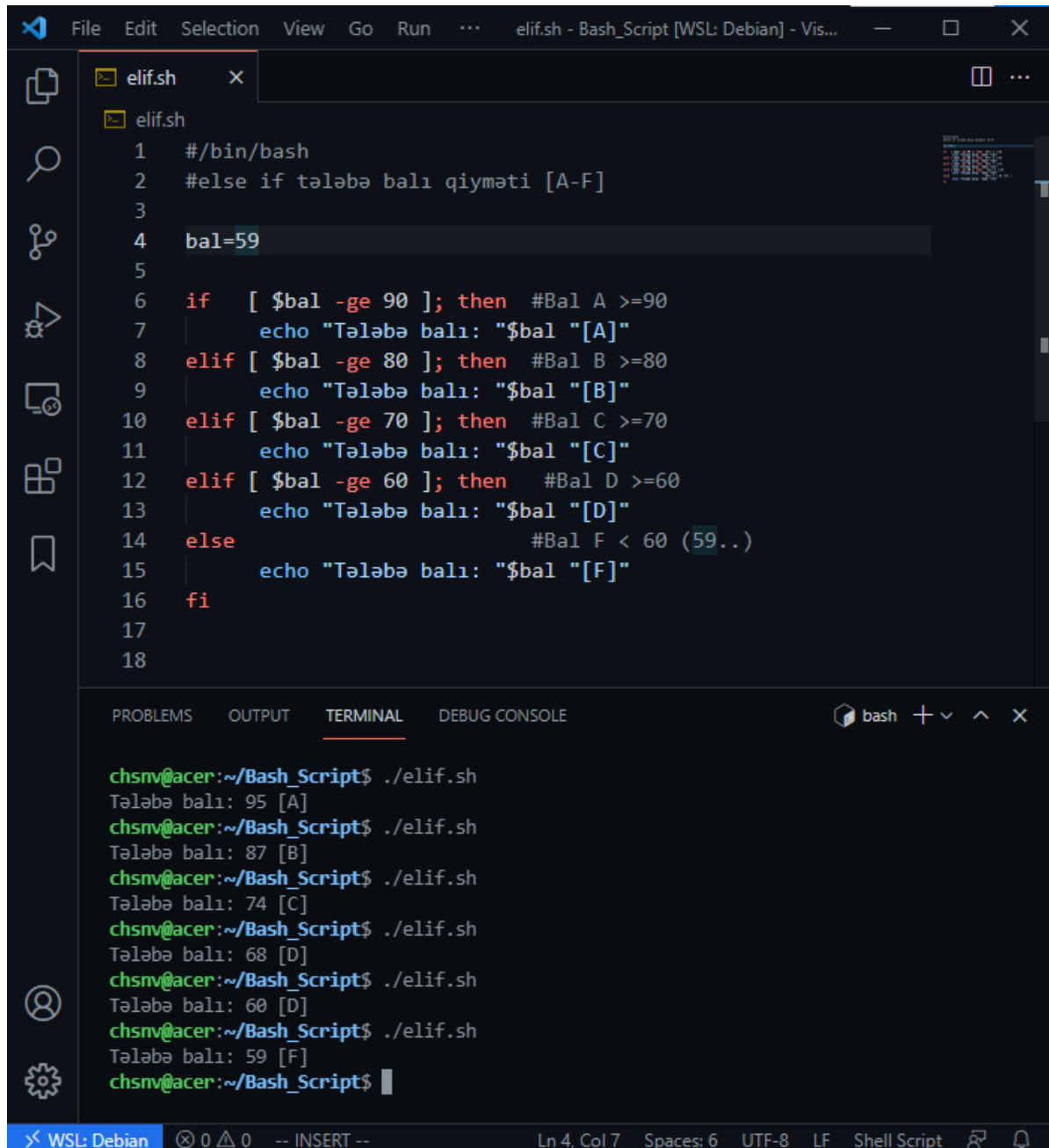
BƏYANAT2

fi

Əgər TEST-ƏMRİ true qaytarsa, the BƏYANAT1 icra olunacaq. Əks təqdirdə, Əgər TEST-ƏMRİ false qaytarsa, BƏYANAT2 icra olunacaq yəni else hissəsi təkcə icra edilir.

if..elif..else Deyimi

Bash if..elif..else strukturu aşağıdakı kimidir:



```
1  #/bin/bash
2  #else if tələbə balı qiyməti [A-F]
3
4  bal=59
5
6  if [ $bal -ge 90 ]; then #Bal A >=90
7      echo "Tələbə balı: "$bal "[A]"
8  elif [ $bal -ge 80 ]; then #Bal B >=80
9      echo "Tələbə balı: "$bal "[B]"
10 elif [ $bal -ge 70 ]; then #Bal C >=70
11     echo "Tələbə balı: "$bal "[C]"
12 elif [ $bal -ge 60 ]; then #Bal D >=60
13     echo "Tələbə balı: "$bal "[D]"
14 else #Bal F < 60 (59..)
15     echo "Tələbə balı: "$bal "[F]"
16 fi
17
18
```

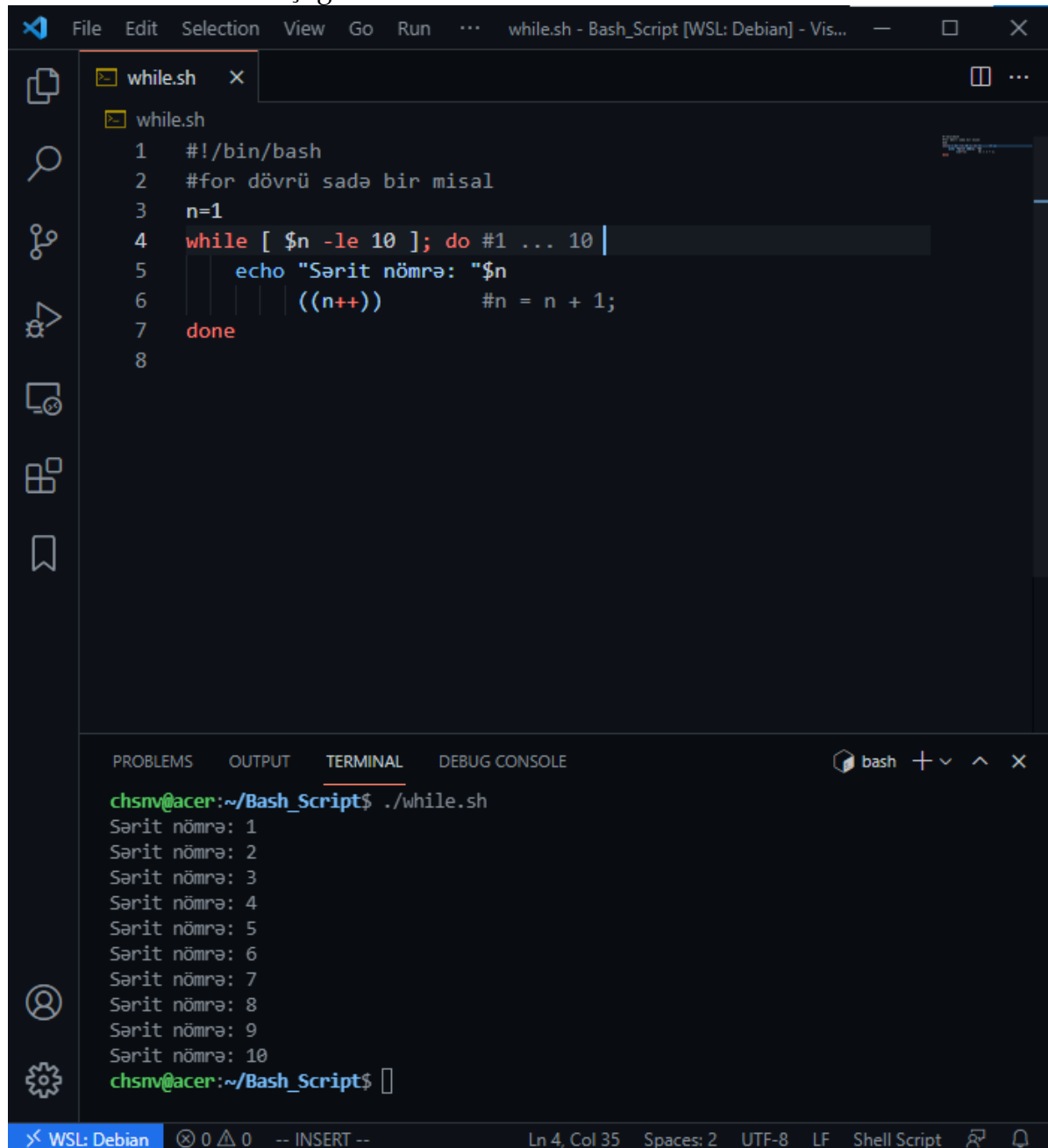
```
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 95 [A]
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 87 [B]
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 74 [C]
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 68 [D]
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 60 [D]
chsnv@acer:~/Bash_Script$ ./elif.sh
Tələbə balı: 59 [F]
chsnv@acer:~/Bash_Script$
```

Qeyd: if TEST-ƏMRİ1 ; then
 BƏYANAT1
elif TEST-ƏMRİ2; then
 BƏYANAT2
else
 BƏYANAT3
fi

Əgər **TEST-ƏMRİ1** *true* dəyəri qaytarsa, **BƏYANAT1** icra olunacaq. Əgər sonra **TEST-ƏMRİ2** *true* qaytararsa, **BƏYANAT2** icra olunacaq. Yox əgər **TEST-ƏMRİ1-2** heç biri *true* qaytarmazsa. Sonda o zaman **BƏYANAT3** icra edilir. Yəni ki, **if** deyimində şərtlər sıralamaya bağlı icra edilir, doğru bir addım olduğu təqdirdə digərlərini yoxlamaq gərəkməz və nəhayət program icra edilir.

Bash **while** dövrü

Bash **while** strukturu aşağıdakı kimidir:



```
File Edit Selection View Go Run ... while.sh - Bash_Script [WSL: Debian] - Vis...
while.sh
1  #!/bin/bash
2  #for dövrü sadə bir misal
3  n=1
4  while [ $n -le 10 ]; do #1 ... 10 |
5      echo "Sərit nömrə: "$n
6      ((n++))             #n = n + 1;
7  done
8

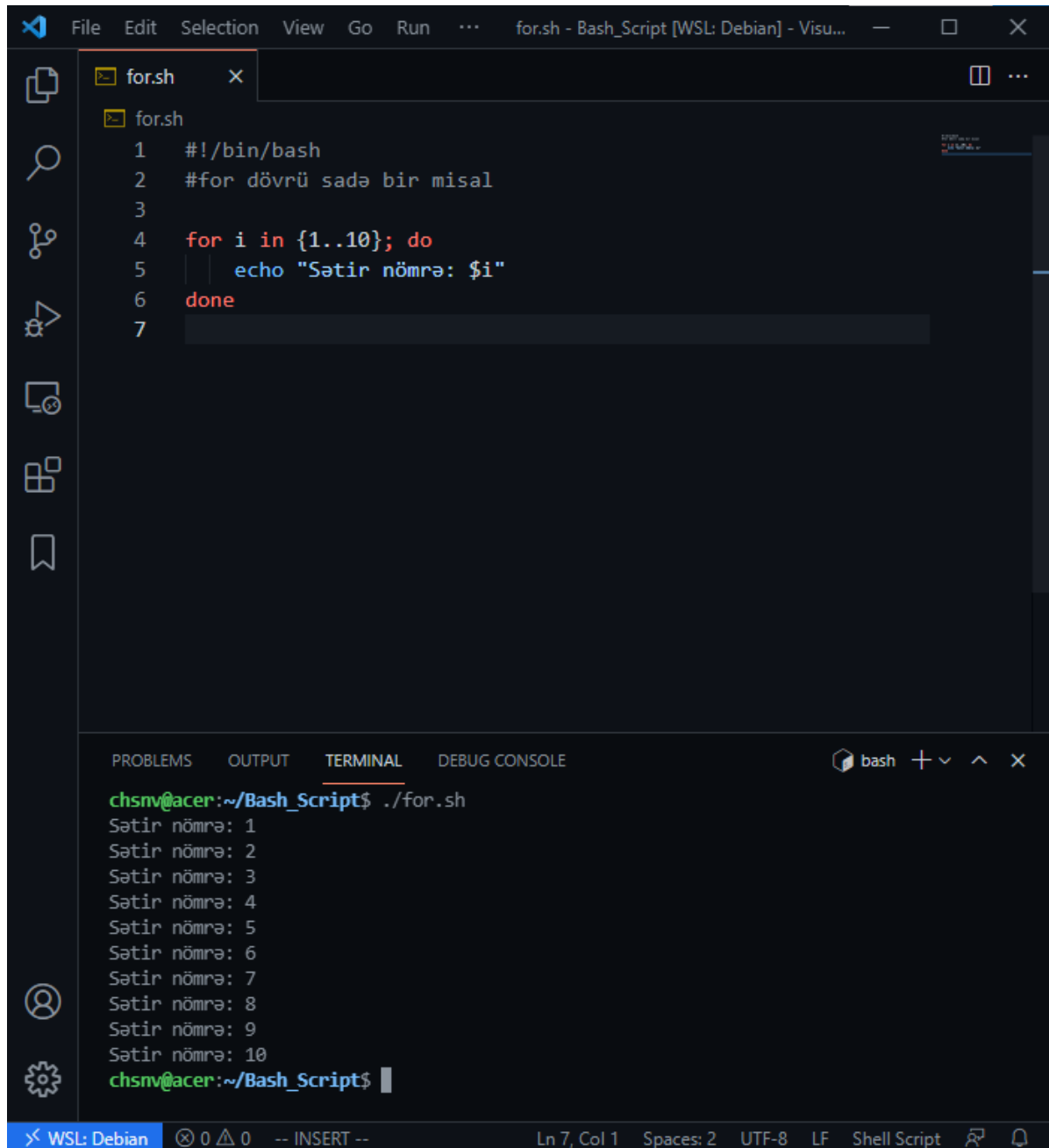
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
chsnv@acer:~/Bash_Script$ ./while.sh
Sərit nömrə: 1
Sərit nömrə: 2
Sərit nömrə: 3
Sərit nömrə: 4
Sərit nömrə: 5
Sərit nömrə: 6
Sərit nömrə: 7
Sərit nömrə: 8
Sərit nömrə: 9
Sərit nömrə: 10
chsnv@acer:~/Bash_Script$
```

Qeyd: **while** [ŞƏRT]
do
[ƏMRLƏR]

Done

Bash Standart **for** dövrü

Bash **for** dövrü strukturu aşağıdakı kimidir:



```
for.sh
1  #!/bin/bash
2  #for dövrü sadə bir misal
3
4  for i in {1..10}; do
5      echo "Sətir nömrə: $i"
6  done
7
```

```
chsnv@acer: ~/Bash_Script$ ./for.sh
Sətir nömrə: 1
Sətir nömrə: 2
Sətir nömrə: 3
Sətir nömrə: 4
Sətir nömrə: 5
Sətir nömrə: 6
Sətir nömrə: 7
Sətir nömrə: 8
Sətir nömrə: 9
Sətir nömrə: 10
chsnv@acer: ~/Bash_Script$
```

Qeyd: **for** item in [SİYAHİ]

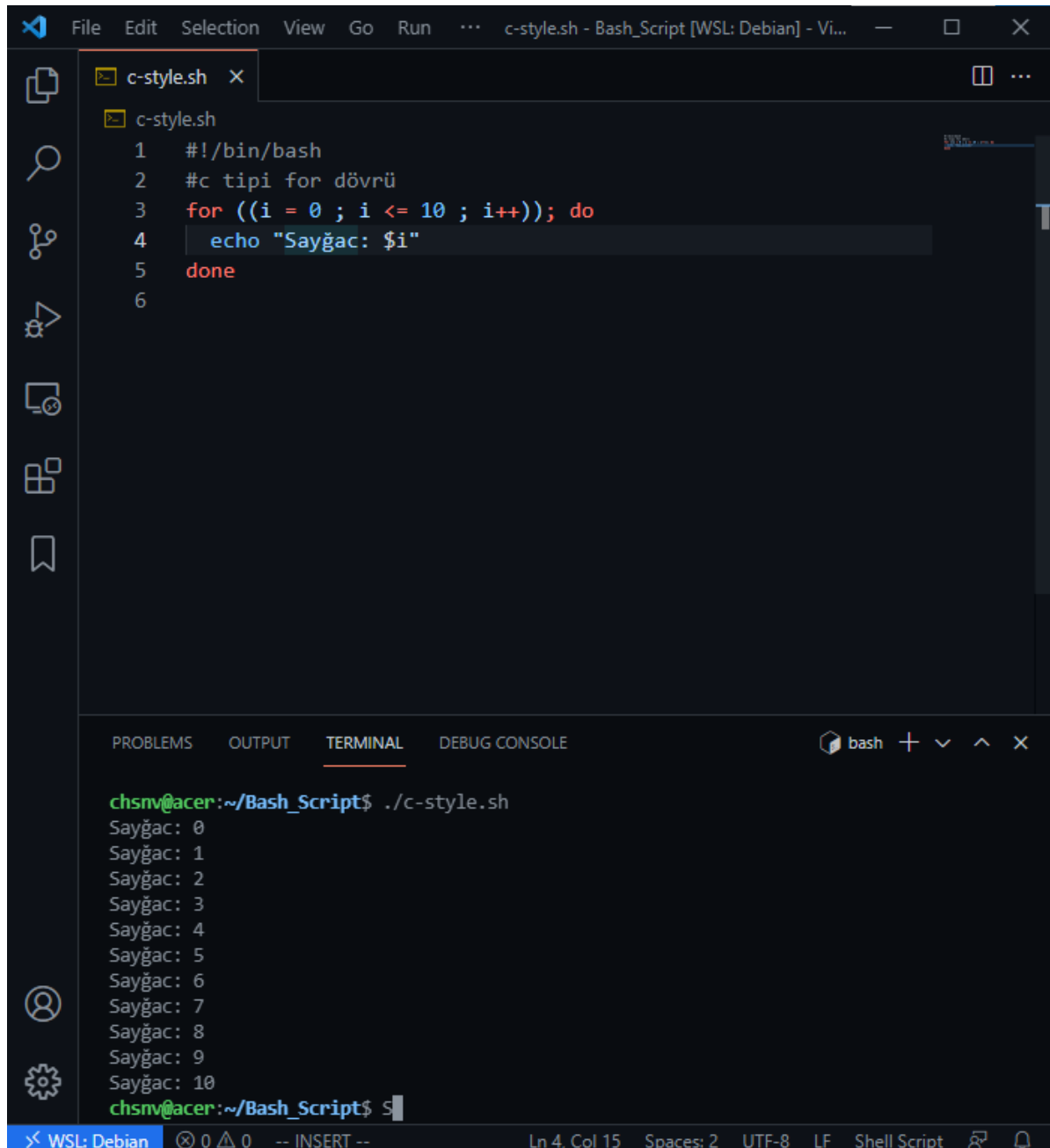
do [ƏMRLƏR]

done

for dövr içərisində müəyyən əmirlərin müəyyən sayda həyata keçməsinə yəni icra edilməsini gerçəkləşdirər.

Bash C-tipi **for** dövrü

Sintaksı c tipi olan **for** dövrünün strukturu aşağıdakı kimidir:



The screenshot shows a VS Code editor window with a file named `c-style.sh`. The script content is as follows:

```
1  #!/bin/bash
2  #c tipi for dövrü
3  for ((i = 0 ; i <= 10 ; i++)); do
4      echo "Sayğac: $i"
5  done
```

Below the editor, the terminal output shows the script being executed, resulting in the following lines:

```
chsnv@acer:~/Bash_Script$ ./c-style.sh
Sayğac: 0
Sayğac: 1
Sayğac: 2
Sayğac: 3
Sayğac: 4
Sayğac: 5
Sayğac: 6
Sayğac: 7
Sayğac: 8
Sayğac: 9
Sayğac: 10
chsnv@acer:~/Bash_Script$
```

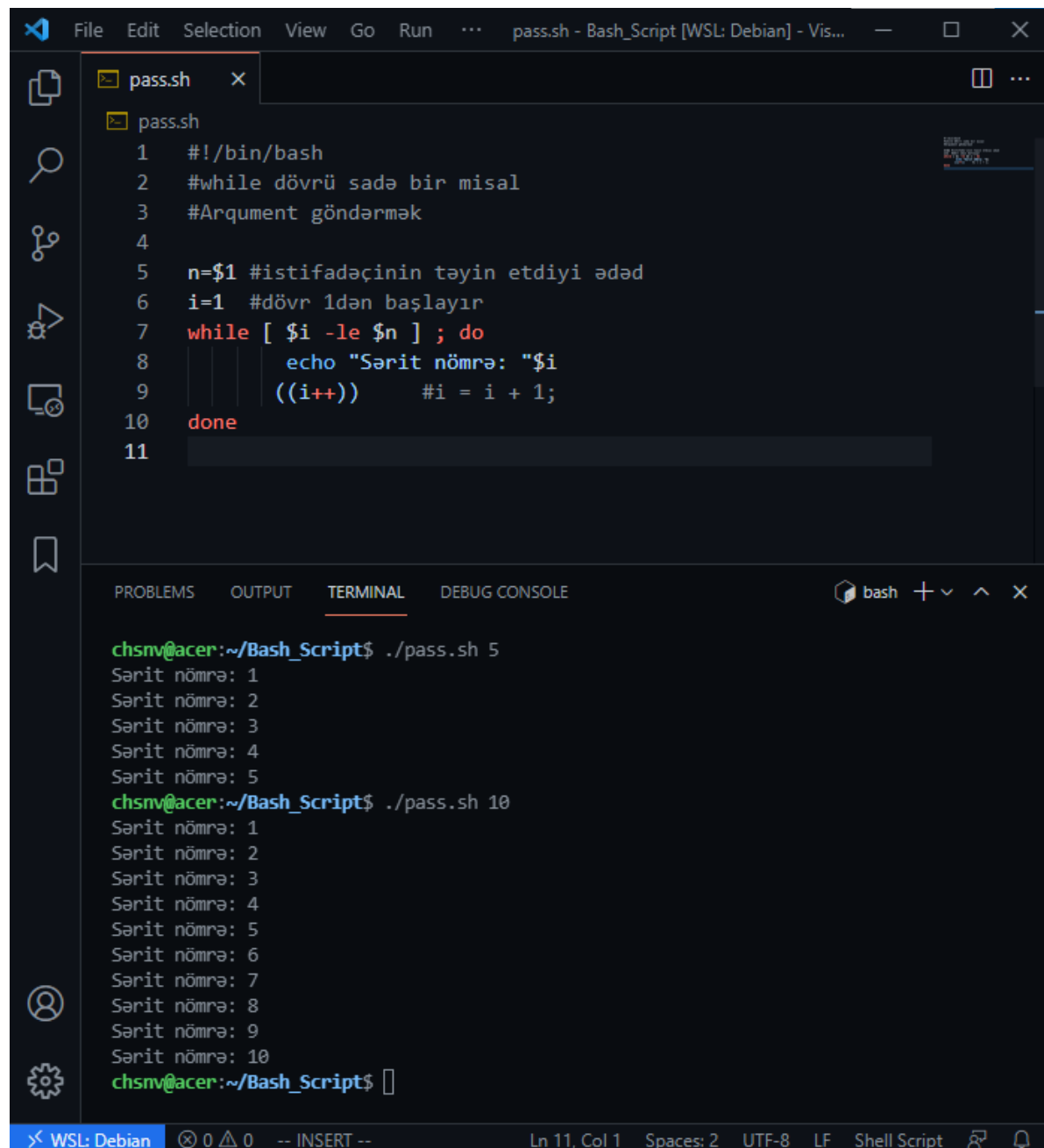
Qeyd: **for** ((BAŞLAMA; TEST; ADDIM)); do
 [ƏMRLƏR]
done

BAŞLAMA hissəsi dövrü başlamadan əvvəl yalnız bir dəfə icra edilir. Sonra **TEST** bölməsi dəyərləndirilir. Yalnız olduqda dövr icra edilməz. **TEST** düzdürsə **for** dövrü daxilində əmrlər toplusu icra edilir. **ADDIM** hissəsi isə hər dəfə yeniləmə edir. Koda baxsaq, **for** dövrü **i = 0** ilə başlasa, hər dəfə **i ≤ 10** olduğu test edilsə və addım **i++**

i-nin hər dəfə bir vahid artırılması olarsa. Dövr şərt **1-10** daxilində **10** dəfə icra edilir. əks təqdirdə isə dövr sonlanır.

Müstəqil dəişən “Passing arguments”

Script üzərindəm məlumat göndərmə parametrləri



```
pass.sh
1  #!/bin/bash
2  #while dövrü sadə bir misal
3  #Argument göndərmək
4
5  n=$1 #istifadəçinin təyin etdiyi ədəd
6  i=1  #dövr 1dən başlayır
7  while [ $i -le $n ] ; do
8      echo "Sərit nömrə: "$i
9      ((i++))      #i = i + 1;
10 done
11
```

```
chsnv@acer:~/Bash_Script$ ./pass.sh 5
Sərit nömrə: 1
Sərit nömrə: 2
Sərit nömrə: 3
Sərit nömrə: 4
Sərit nömrə: 5
chsnv@acer:~/Bash_Script$ ./pass.sh 10
Sərit nömrə: 1
Sərit nömrə: 2
Sərit nömrə: 3
Sərit nömrə: 4
Sərit nömrə: 5
Sərit nömrə: 6
Sərit nömrə: 7
Sərit nömrə: 8
Sərit nömrə: 9
Sərit nömrə: 10
chsnv@acer:~/Bash_Script$
```

Qeyd: Bash qabığı üzərindən program script ica edilərkən istifadəçi tərəfinbən bir əmr, seçim edilməsi yəni scriptə məlumat göndərmək üçün itifadə edilir:

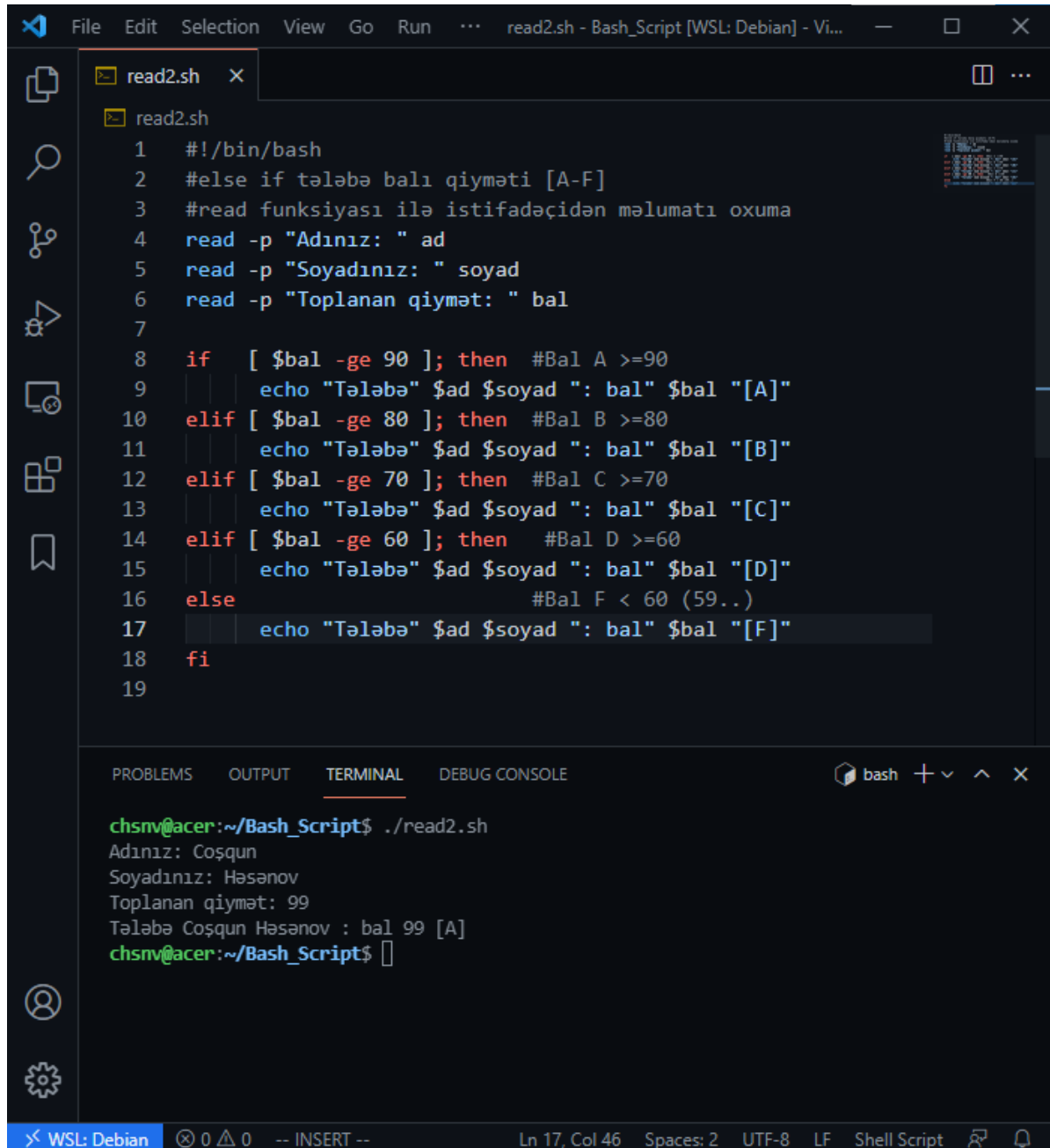
\$0

\$1 ilk parametrlər

\$2-\$9 ardıcılığı ilə bu davam edər.

Read əmri

Məlumat daxil edilməsi (*kalviyatura*) üzərindən məlumatın işlənməsi



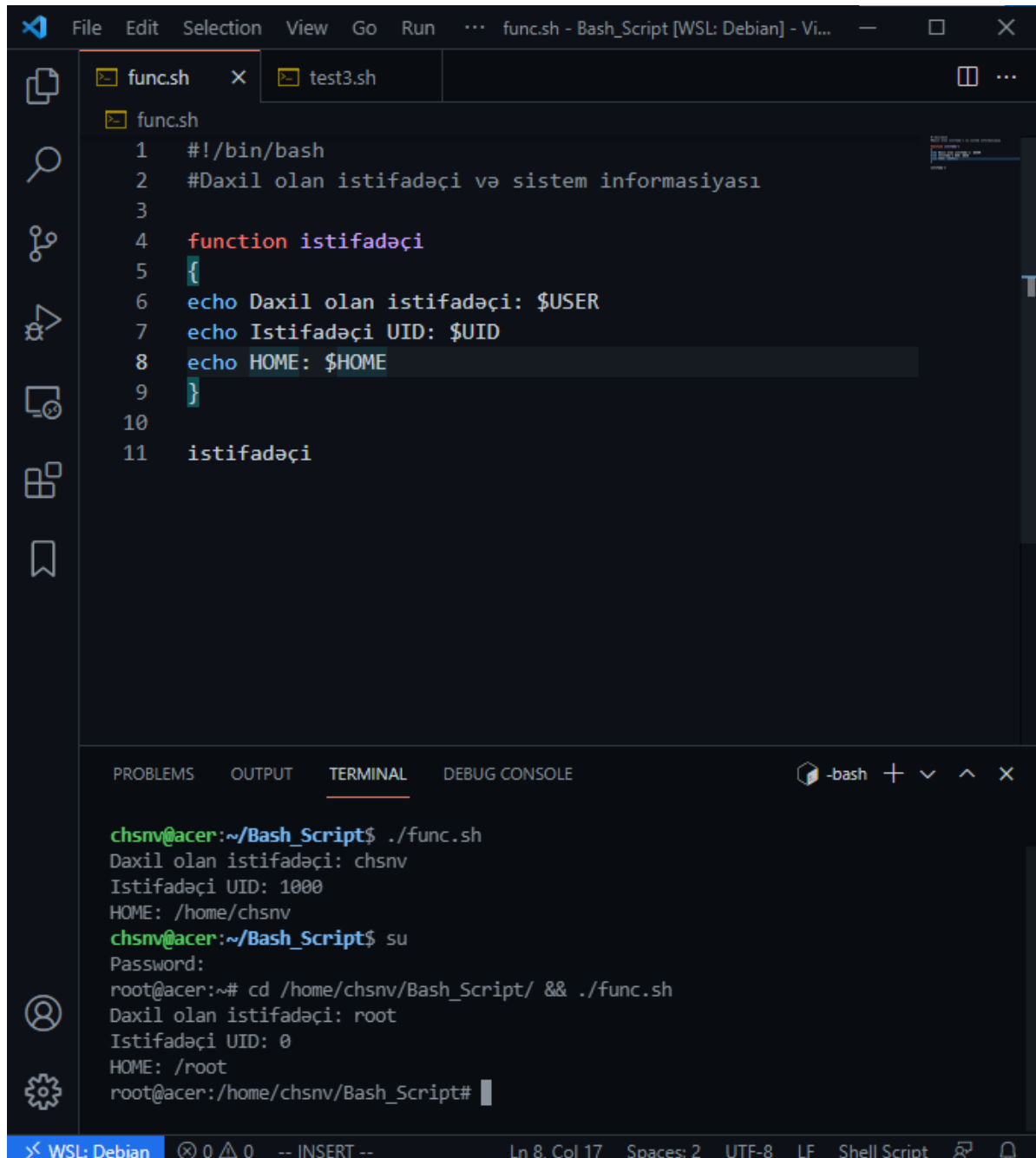
```
1  #!/bin/bash
2  #else if tələbə balı qiyməti [A-F]
3  #read funksiyası ilə istifadəçidən məlumatı oxuma
4  read -p "Adınız: " ad
5  read -p "Soyadınız: " soyad
6  read -p "Toplanan qiymət: " bal
7
8  if [ $bal -ge 90 ]; then #Bal A >=90
9      echo "Tələbə" $ad $soyad ": bal" $bal "[A]"
10 elif [ $bal -ge 80 ]; then #Bal B >=80
11     echo "Tələbə" $ad $soyad ": bal" $bal "[B]"
12 elif [ $bal -ge 70 ]; then #Bal C >=70
13     echo "Tələbə" $ad $soyad ": bal" $bal "[C]"
14 elif [ $bal -ge 60 ]; then #Bal D >=60
15     echo "Tələbə" $ad $soyad ": bal" $bal "[D]"
16 else #Bal F < 60 (59..)
17     echo "Tələbə" $ad $soyad ": bal" $bal "[F]"
18 fi
19
```

```
chsnv@acer:~/Bash_Script$ ./read2.sh
Adınız: Coşqun
Soyadınız: Həsənov
Toplanan qiymət: 99
Tələbə Coşqun Həsənov : bal 99 [A]
chsnv@acer:~/Bash_Script$
```

Qeyd: **read** əmrinin ala biləcəri yəni **-help** ilə CLI üzərindən təyin edə bilərsiniz. **-t 5** parametri ilə istifadəşi **5** saniyə zaman kəsində məlumatı daxil edilməz isə sonrakı sətir icra edilir və nəhatət script sonlanır.

Funksiya "Functions"

Funksiya iki ayrı formatda `funksiya_adı () {..}` və ya `function funksiya_adı {...}`



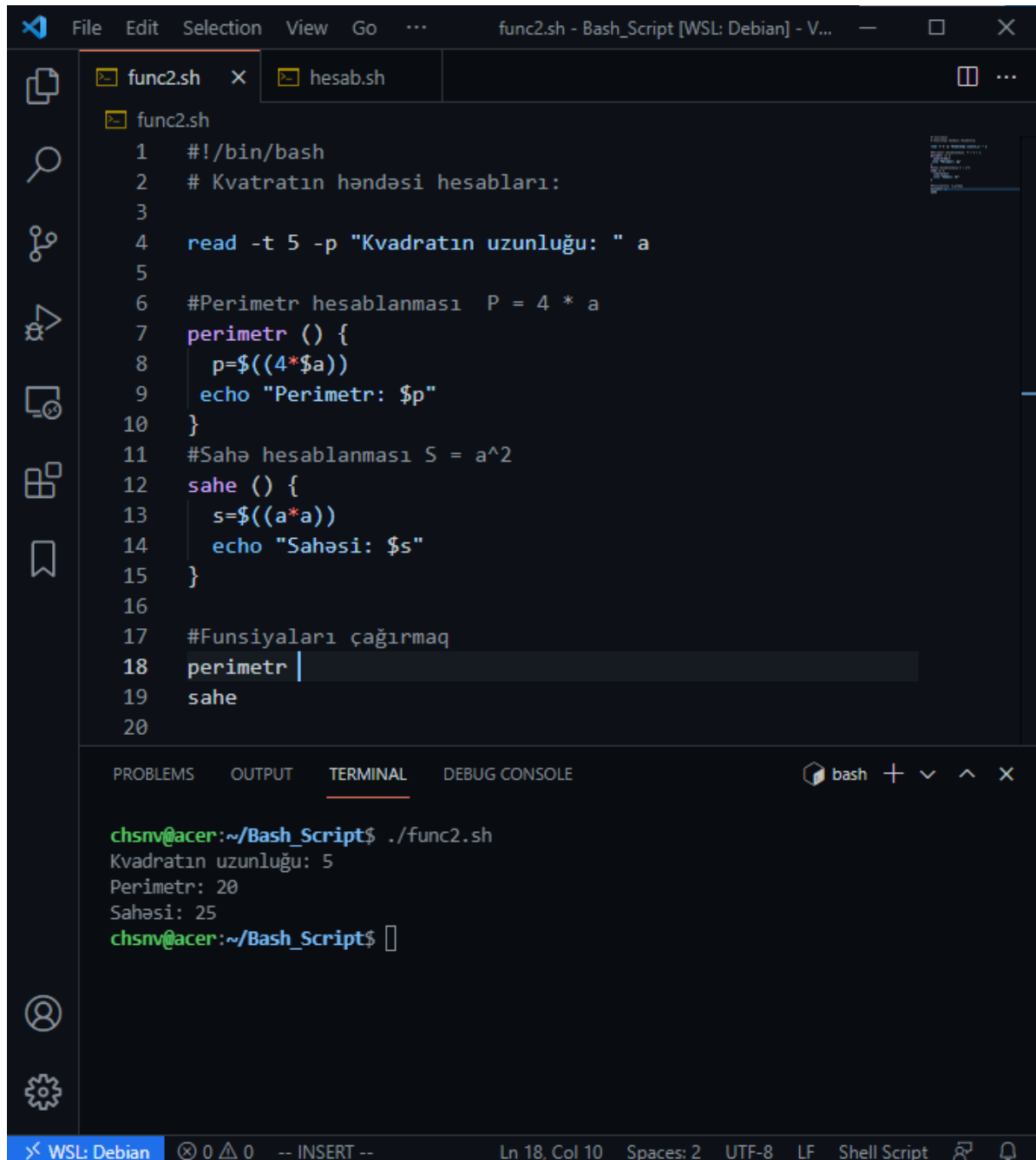
```
func.sh
1  #!/bin/bash
2  #Daxil olan istifadəçi və sistem informasiyası
3
4  function istifadəçi
5  {
6  echo Daxil olan istifadəçi: $USER
7  echo İstifadəçi UID: $UID
8  echo HOME: $HOME
9  }
10
11  istifadəçi

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
chsnv@acer:~/Bash_Script$ ./func.sh
Daxil olan istifadəçi: chsnv
İstifadəçi UID: 1000
HOME: /home/chsnv
chsnv@acer:~/Bash_Script$ su
Password:
root@acer:~# cd /home/chsnv/Bash_Script/ && ./func.sh
Daxil olan istifadəçi: root
İstifadəçi UID: 0
HOME: /root
root@acer:/home/chsnv/Bash_Script#
```

Qeyd: Mötərizə daxilindəki əmrlər `{}` funksiyanın gövdəsini təşkil edir. Tək sətirli funksiyalarda əmdən sonra `;` nöqtəli vergül istifadə edlər. Funksiya adını yadda qalan və açıqlayıcı şəkildə saxlamaq daha səmərəlidir. Funksiyanın çağırılması zamanı sadəcə adı qeyd edilir.

Funksiya

Script hazırlamaqla yanaşı qabıq "shell" mühitində funksiyaadan istidadə edilə bilər.



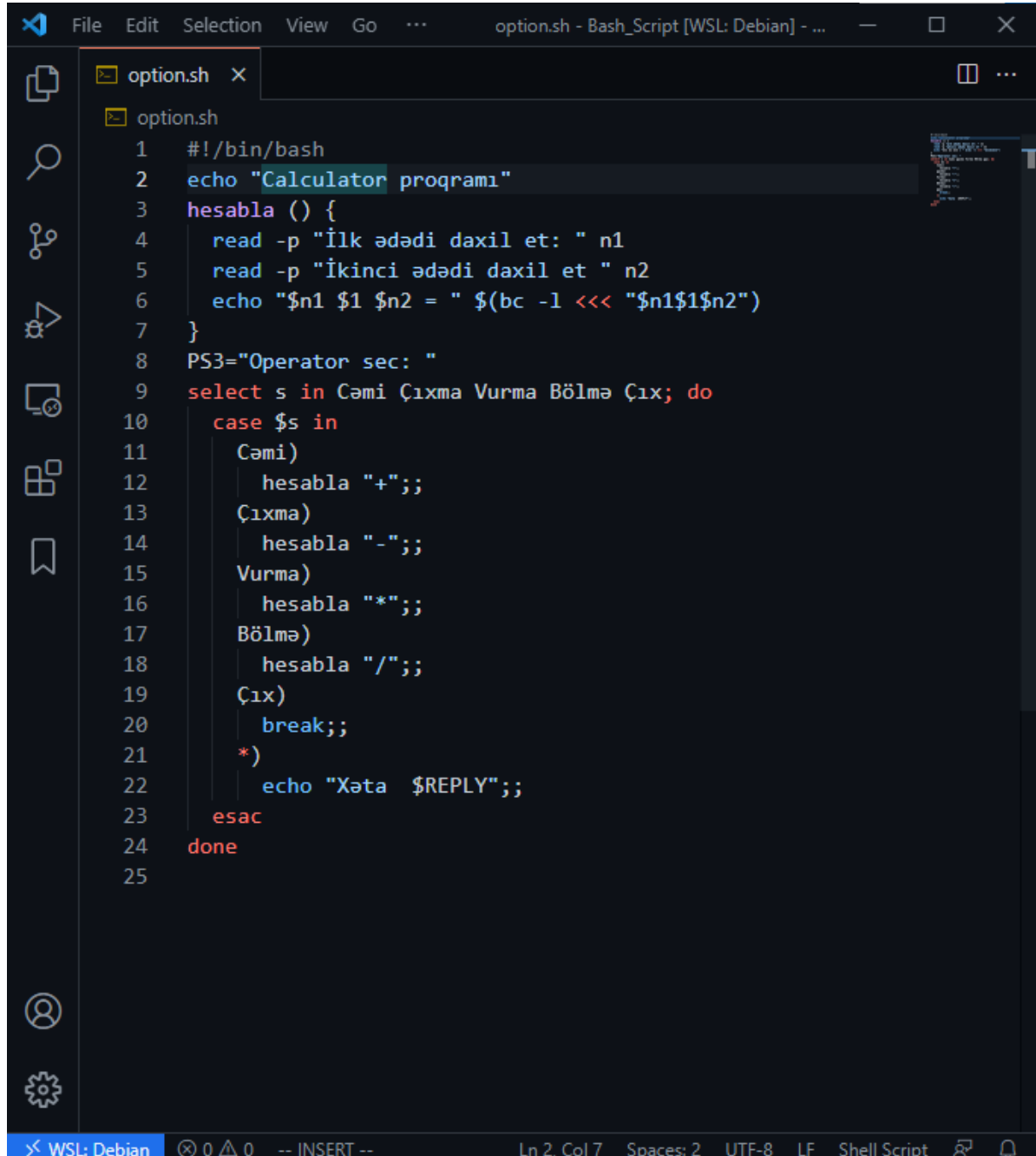
```
func2.sh
1  #!/bin/bash
2  # Kvadratın həndəsi hesabları:
3
4  read -t 5 -p "Kvadratın uzunluğu: " a
5
6  #Perimetr hesablanması P = 4 * a
7  perimetr () {
8      p=$((4*$a))
9      echo "Perimetr: $p"
10 }
11 #Sahə hesablanması S = a^2
12 sahe () {
13     s=$((a*a))
14     echo "Sahəsi: $s"
15 }
16
17 #Funksiyaları çağırmaq
18 perimetr
19 sahe
20
```

```
chsnv@acer:~/Bash_Script$ ./func2.sh
Kvadratın uzunluğu: 5
Perimetr: 20
Sahəsi: 25
chsnv@acer:~/Bash_Script$
```

Qeyd: Qabıq mühitində funksiyaları görmək üçün **declare -F** əmri istifadə edilir.

case Bəyanatı

Bash case bəyanatı strukturu aşağıdakı kimidir:



```
1  #!/bin/bash
2  echo "Calculator programı"
3  hesabla () {
4      read -p "İlk ədədi daxil et: " n1
5      read -p "İkinci ədədi daxil et " n2
6      echo "$n1 $1 $n2 = " $(bc -l <<< "$n1$1$n2")
7  }
8  PS3="Operator sec: "
9  select s in Cəmi Çıxma Vurma Bölmə Çıx; do
10     case $s in
11         Cəmi)
12             hesabla "+";;
13         Çıxma)
14             hesabla "-";;
15         Vurma)
16             hesabla "*";;
17         Bölmə)
18             hesabla "/";;
19         Çıx)
20             break;;
21         *)
22             echo "Xəta $REPLY";;
23     esac
24 done
25
```

Qeyd: case ifadə in

proses1)

BƏYANAT1

BƏYANAT2 ;;

...

*)

Esac

Bash **Select** Menu

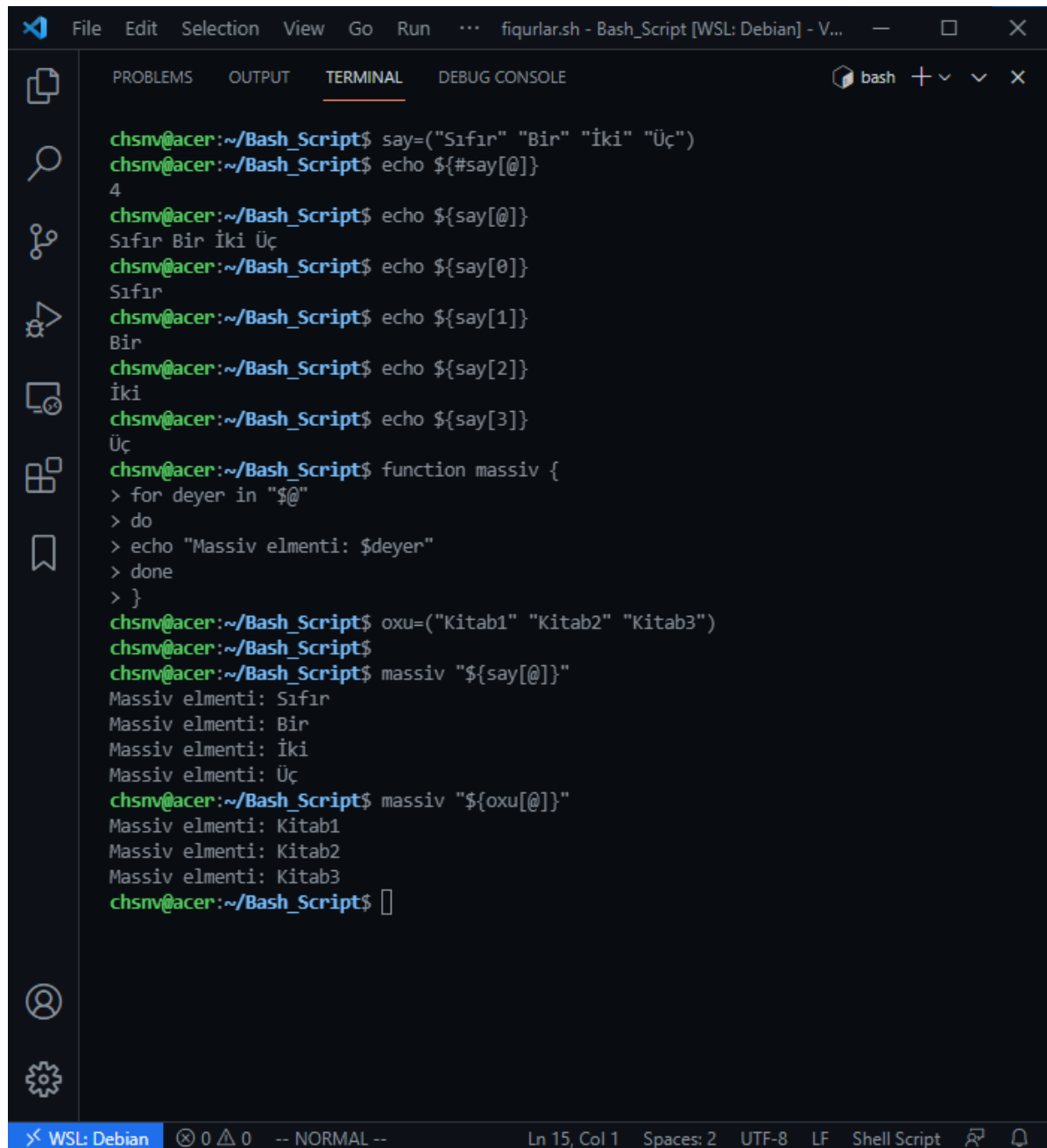
Select Menyu qurucusu for dövrü ilə eyni bir sturukrura malikdir.

```
option.sh - Bash_Script [WSL: Debian] - ...  
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE  
bash + v v X  
chsnv@acer: ~/Bash_Script$ ./option.sh  
Calculator programı  
1) Cəmi  
2) Çıxma  
3) Vurma  
4) Bölmə  
5) Çıx  
Operator sec: 1  
İlk ədədi daxil et: 5  
İkinci ədədi daxil et 7  
5 + 7 = 12  
Operator sec: 2  
İlk ədədi daxil et: 8  
İkinci ədədi daxil et 9  
8 - 9 = -1  
Operator sec: 3  
İlk ədədi daxil et: 5  
İkinci ədədi daxil et 2  
5 * 2 = 10  
Operator sec: 4  
İlk ədədi daxil et: 3  
İkinci ədədi daxil et 2  
3 / 2 = 1.50000000000000000000  
Operator sec: 5  
chsnv@acer: ~/Bash_Script$
```

```
Qeyd: select bənd in [SİYAHII]
      do [ƏMRLƏR]
      done
```


Massivlər "Arrays"

Bash massiv sturukturu aşağıdakılar kimidir:



```
chsnv@acer:~/Bash_Script$ say=("Sıfır" "Bir" "İki" "Üç")
chsnv@acer:~/Bash_Script$ echo ${#say[@]}
4
chsnv@acer:~/Bash_Script$ echo ${say[@]}
Sıfır Bir İki Üç
chsnv@acer:~/Bash_Script$ echo ${say[0]}
Sıfır
chsnv@acer:~/Bash_Script$ echo ${say[1]}
Bir
chsnv@acer:~/Bash_Script$ echo ${say[2]}
İki
chsnv@acer:~/Bash_Script$ echo ${say[3]}
Üç
chsnv@acer:~/Bash_Script$ function massiv {
> for deyer in "$@"
> do
> echo "Massiv elmenti: $deyer"
> done
> }
chsnv@acer:~/Bash_Script$ oxu=("Kitab1" "Kitab2" "Kitab3")
chsnv@acer:~/Bash_Script$
chsnv@acer:~/Bash_Script$ massiv "${say[@]}"
Massiv elmenti: Sıfır
Massiv elmenti: Bir
Massiv elmenti: İki
Massiv elmenti: Üç
chsnv@acer:~/Bash_Script$ massiv "${oxu[@]}"
Massiv elmenti: Kitab1
Massiv elmenti: Kitab2
Massiv elmenti: Kitab3
chsnv@acer:~/Bash_Script$
```

Qeyd: Məlumatın sonradan əlavə edilməsi += silinməsi unset və s açar söz və operatorlar istifadə edilir.

Funksiya **Select** və **Case**

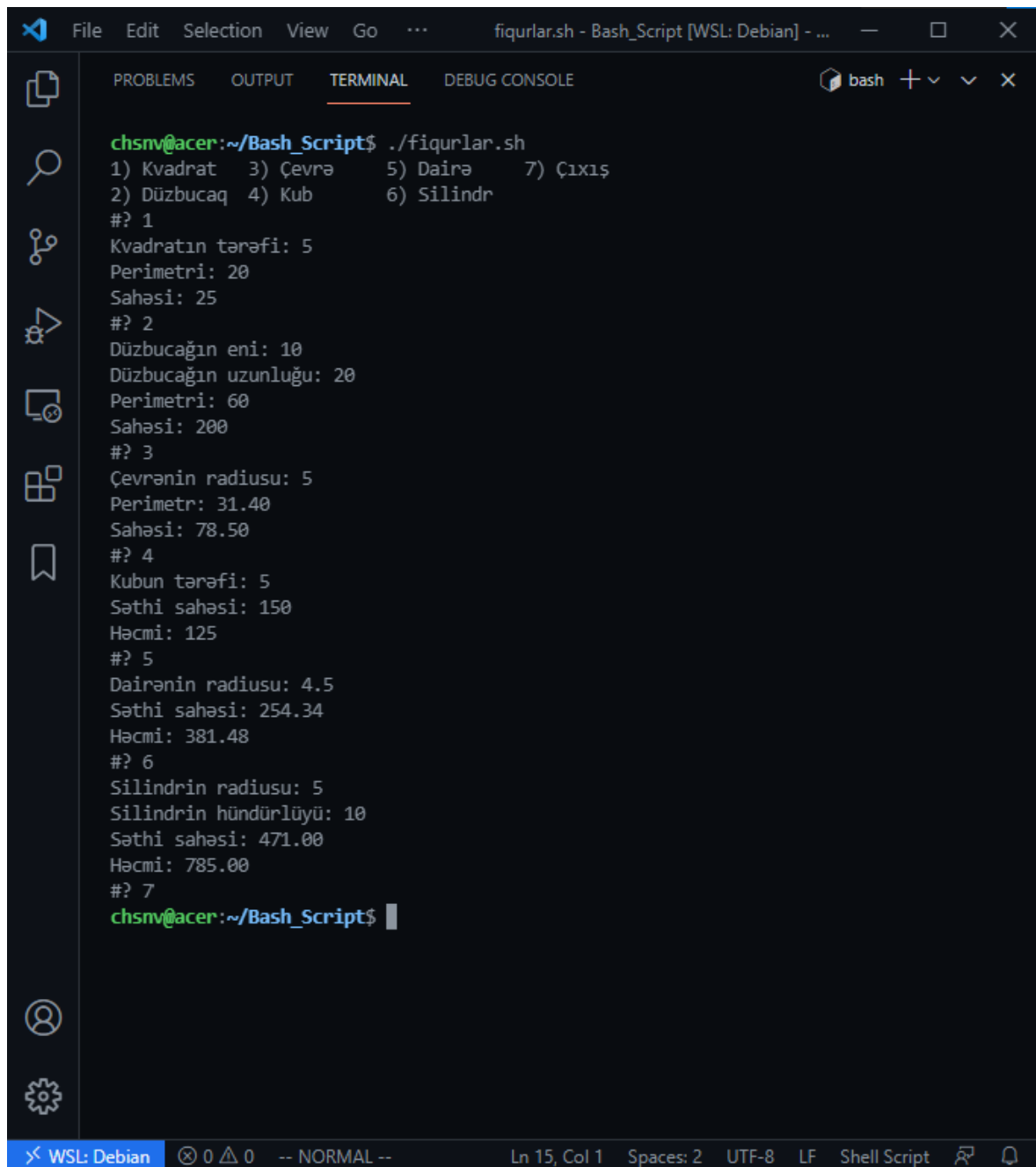
```
fiqurlar.sh
1  #!/bin/bash
2  # 'two-dimensional' 2-öçülü fiqurlar Sahə^2
3  # Kvadratın həndəsi hesabları
4  # Düzbucaqlının həndəsi hesabları
5  # Çevrənin həndəsi hesabları
6  # 'three-dimensional' 3-öçülü fiqurlar Səthi^2, Həcmi^3
7  # Kubun həndəsi hesabları
8  # Dairənin (Sfera/Kürə) həndəsi hesabları
9  # Silindrin həndəsi hesabları
10
11 # Kvadrat perimetr hesablanması P = 4 * a
12 perimetr_kvadrat () {
13     let cavab="4 * a"
14     return $cavab
15 }
16 # Kvadrat sahə hesablanması S = a^2
17 sahe_kvadrat () {
18     let cavab="a * a"
19     return $cavab
20 }
21 # Düzbucaq perimetr hesablanması P = 2 *(e + u)
22 perimetr_duzbucaq () {
23     let cavab="2 * (e + u)"
24     return $cavab
25 }
26 # Düzbucaq sahə hesablanması S = e * u
27 sahe_duzbucaq () {
28     let cavab="e * u"
29     return $cavab
30 }
31 # Çevrə perimetr hesablanması P = 2 * pi * r
32 function perimetr_cevre {
33     cavab=$(bc -l <<< "2 * 3.14 * $r")
34     echo "Perimetr: $cavab"
35 }
36 # Çevrə sahə hesablanması S = 3 * r^2
37 function sahe_cevre {
38     cavab=$(bc -l <<< "scale=1; 3.14 * $r * $r")
39     echo "Sahəsi: $cavab"
40 }
41 # Kubun səthi sahə hesablanması S = 6 * a^2
42 sethi_sahe_kub (){
43     let cavab="6 * a * a"
44     return $cavab
45 }
46 # Kubun həcmi hesablanması V = a^3
47 hecm_kub () {
48     let cavab="a * a * a"
49     return $cavab
50 }
51 # Dairənin səthi sahə hesablanması S = 4 * pi * r^2
52 function sethi_sahe_daire {
53     cavab=$(bc -l <<< "scale=2; 4 * 3.14 * $r * $r")
54     echo "Səthi sahəsi: $cavab"
55 }
56 # Dairənin həcmi hesablanması V = 4/3 * pi * r^3
57 function hecm_daire () {
58     cavab=$(bc -l <<< "scale=2; 4 * (3.14 * $r * $r * $r) / 3")
59     echo "Həcmi: $cavab"
60 }
61 # Silindrin səthi sahə hesablanması S = 2 * pi * r * (H + r)
62 function sethi_sahe_silindr {
63     cavab=$(bc -l <<< "2 * 3.14 * $r * ($h + $r)")
64 }
```

Funksiya **Select** və **Case** davamı

```
fiqurlar.sh
61 # Silindrin səthi sahə hesablanması  $S = 2 * \pi * r * (h + r)$ 
62 function sethi_sahe_silindr {
63     cavab=$(bc -l <<< "2 * 3.14 * $r * ($h + $r)")
64     echo "Səthi sahəsi: $cavab"
65 }
66 # Silindrin həcmi hesablanması  $V = \pi * r^2 * h$ 
67 function hecm_silindr {
68     cavab=$(bc -l <<< "3.14 * $r * $r * $h")
69     echo "Həcmi: $cavab"
70 }
71
72 # Menu hissəsi
73 select s in Kvadrat Düzbucaq Çevrə Kub Dairə Silindr Çıxış; do
74     case $s in
75         Kvadrat)
76             read -p "Kvadratın tərəfi: " a
77             perimetr_kvadrat
78             echo "Perimetri: $?"
79             sahe_kvadrat
80             echo "Sahəsi: $?"
81             ;;
82         Düzbucaq)
83             read -p "Düzbucağın eni: " e
84             read -p "Düzbucağın uzunluğu: " u
85             perimetr_duzbucaq
86             echo "Perimetri: $?"
87             sahe_duzbucaq
88             echo "Sahəsi: $?"
89             ;;
90         Çevrə)
91             read -p "Çevrənin radiusu: " r
92             perimetr_cevre
93             sahe_cevre
94             ;;
95         Kub)
96             read -p "Kubun tərəfi: " a
97             sethi_sahe_kub
98             echo "Səthi sahəsi: $?"
99             hecm_kub
100             echo "Həcmi: $?"
101             ;;
102         Dairə)
103             read -p "Dairənin radiusu: " r
104             sethi_sahe_daire
105             hecm_daire
106             ;;
107         Silindr)
108             read -p "Silindrin radiusu: " r
109             read -p "Silindrin hündürlüyü: " h
110             sethi_sahe_silindr
111             hecm_silindr
112             ;;
113         Çıxış)
114             break
115             ;;
116         *)
117             echo "Xəta $REPLY"
118             ;;
119     esac
120 done
121
```

Funksiya **Select** və **Case** son

Həndəsi fiqurlar və riyazi hesablamalar nəticə

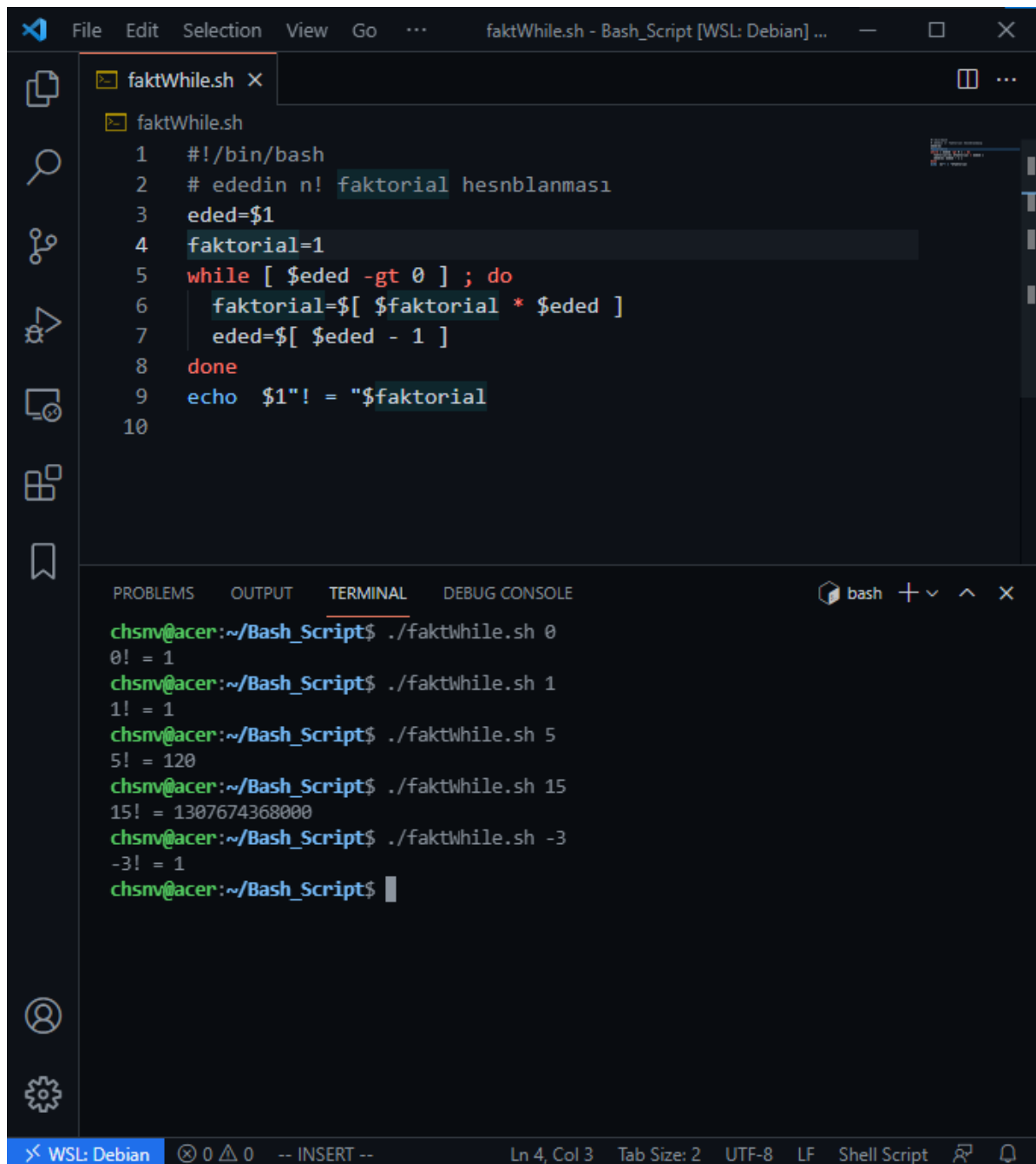


```
File Edit Selection View Go ... fiqurlar.sh - Bash_Script [WSL: Debian] - ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
chsnv@acer:~/Bash_Script$ ./fiqurlar.sh
1) Kvadrat 3) Çevrə 5) Dairə 7) Çıxış
2) Düzbucaq 4) Kub 6) Silindr
#? 1
Kvadratın tərəfi: 5
Perimetri: 20
Sahəsi: 25
#? 2
Düzbucağın eni: 10
Düzbucağın uzunluğu: 20
Perimetri: 60
Sahəsi: 200
#? 3
Çevrənin radiusu: 5
Perimetr: 31.40
Sahəsi: 78.50
#? 4
Kubun tərəfi: 5
Səthi sahəsi: 150
Həcmi: 125
#? 5
Dairənin radiusu: 4.5
Səthi sahəsi: 254.34
Həcmi: 381.48
#? 6
Silindrin radiusu: 5
Silindrin hündürlüyü: 10
Səthi sahəsi: 471.00
Həcmi: 785.00
#? 7
chsnv@acer:~/Bash_Script$
```

Qeyd: Script üzərindən məlumatın işlənməsi, funksiyanın tətbiq edilmə methodlarının hər iki tipindən istifadə etmişəm. Əgər real ədədlər ilə hesablamalar etmək bash ilə qarışıq hal ala bilər. Həll yolu **bc** yəni “bash calculator” terminal hesablama proqramıdır. Proqram özündə bəzi limitlər mövcuddur.

Faktorial

while ilə $n!$ faktorial hesablama nümunə

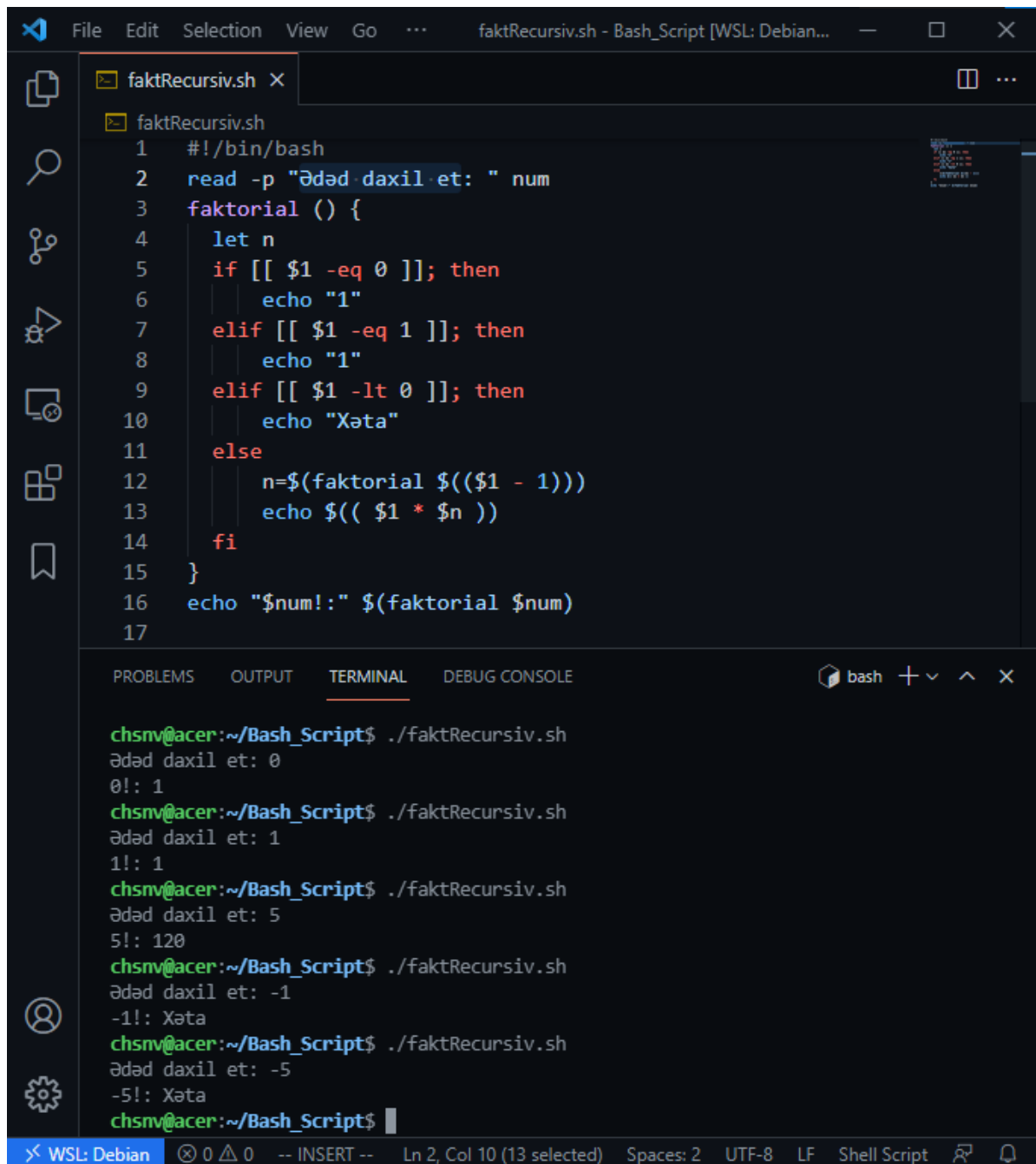


```
faktWhile.sh
1  #!/bin/bash
2  # ededin n! faktorial hesnblanması
3  eded=$1
4  faktorial=1
5  while [ $eded -gt 0 ] ; do
6      faktorial=$(( faktorial * eded ))
7      eded=$(( eded - 1 ))
8  done
9  echo "$1!" = "$faktorial"
10
```

```
chsnv@acer:~/Bash_Script$ ./faktWhile.sh 0
0! = 1
chsnv@acer:~/Bash_Script$ ./faktWhile.sh 1
1! = 1
chsnv@acer:~/Bash_Script$ ./faktWhile.sh 5
5! = 120
chsnv@acer:~/Bash_Script$ ./faktWhile.sh 15
15! = 1307674368000
chsnv@acer:~/Bash_Script$ ./faktWhile.sh -3
-3! = 1
chsnv@acer:~/Bash_Script$
```

Faktorial

Rekursiv funksiya ilə $n!$ ədədin hesablanması

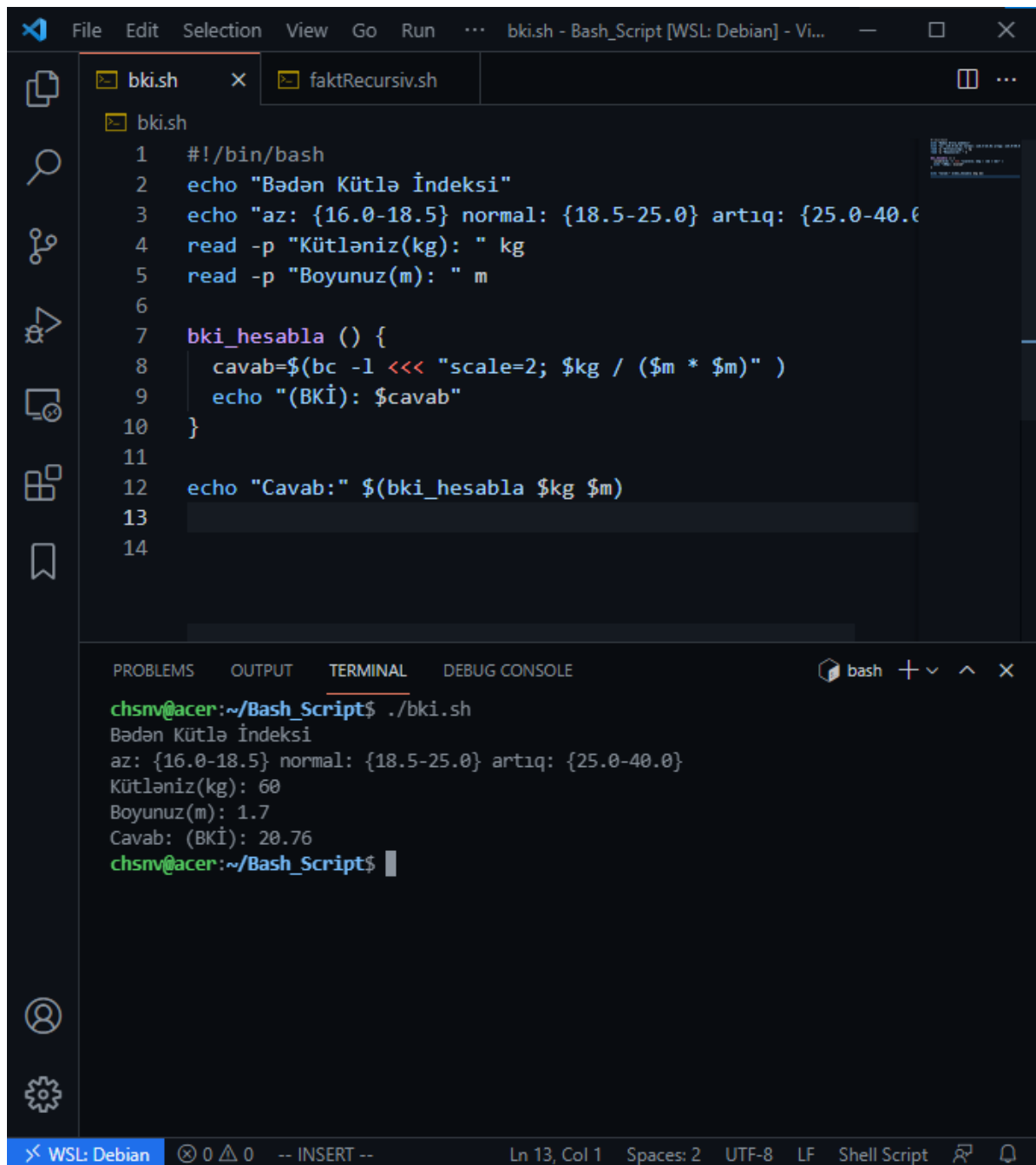


```
faktRekursiv.sh
1  #!/bin/bash
2  read -p "Ədəd daxil et: " num
3  faktorial () {
4      let n
5      if [[ $1 -eq 0 ]]; then
6          echo "1"
7      elif [[ $1 -eq 1 ]]; then
8          echo "1"
9      elif [[ $1 -lt 0 ]]; then
10         echo "Xəta"
11     else
12         n=$(faktorial $(( $1 - 1 )) )
13         echo $(( $1 * $n ))
14     fi
15 }
16 echo "$num!:" $(faktorial $num)
17
```

```
chsnv@acer:~/Bash_Script$ ./faktRekursiv.sh
Ədəd daxil et: 0
0!: 1
chsnv@acer:~/Bash_Script$ ./faktRekursiv.sh
Ədəd daxil et: 1
1!: 1
chsnv@acer:~/Bash_Script$ ./faktRekursiv.sh
Ədəd daxil et: 5
5!: 120
chsnv@acer:~/Bash_Script$ ./faktRekursiv.sh
Ədəd daxil et: -1
-1!: Xəta
chsnv@acer:~/Bash_Script$ ./faktRekursiv.sh
Ədəd daxil et: -5
-5!: Xəta
chsnv@acer:~/Bash_Script$
```

BKİ

Bədən kütlə indeksi nümunə

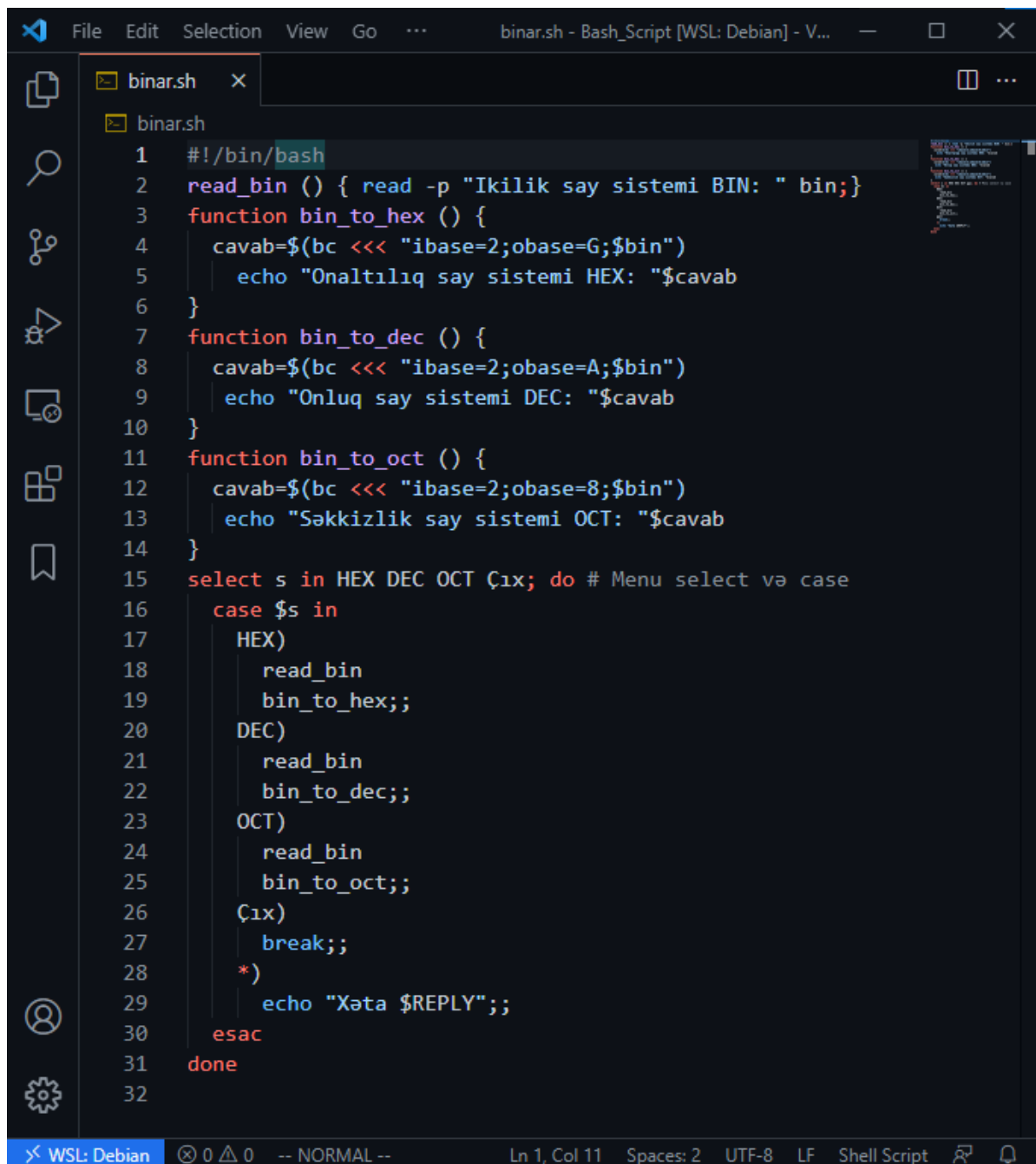


```
File Edit Selection View Go Run ... bki.sh - Bash_Script [WSL: Debian] - Vi...
bki.sh
1  #!/bin/bash
2  echo "Bədən Kütlə İndeksi"
3  echo "az: {16.0-18.5} normal: {18.5-25.0} artıq: {25.0-40.0}"
4  read -p "Kütləniz(kg): " kg
5  read -p "Boyunuz(m): " m
6
7  bki_hesabla () {
8      cavab=$(bc -l <<< "scale=2; $kg / ($m * $m)" )
9      echo "(BKİ): $cavab"
10 }
11
12 echo "Cavab:" $(bki_hesabla $kg $m)
13
14

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
chsnv@acer:~/Bash_Script$ ./bki.sh
Bədən Kütlə İndeksi
az: {16.0-18.5} normal: {18.5-25.0} artıq: {25.0-40.0}
Kütləniz(kg): 60
Boyunuz(m): 1.7
Cavab: (BKİ): 20.76
chsnv@acer:~/Bash_Script$
```

Binar "Binary"

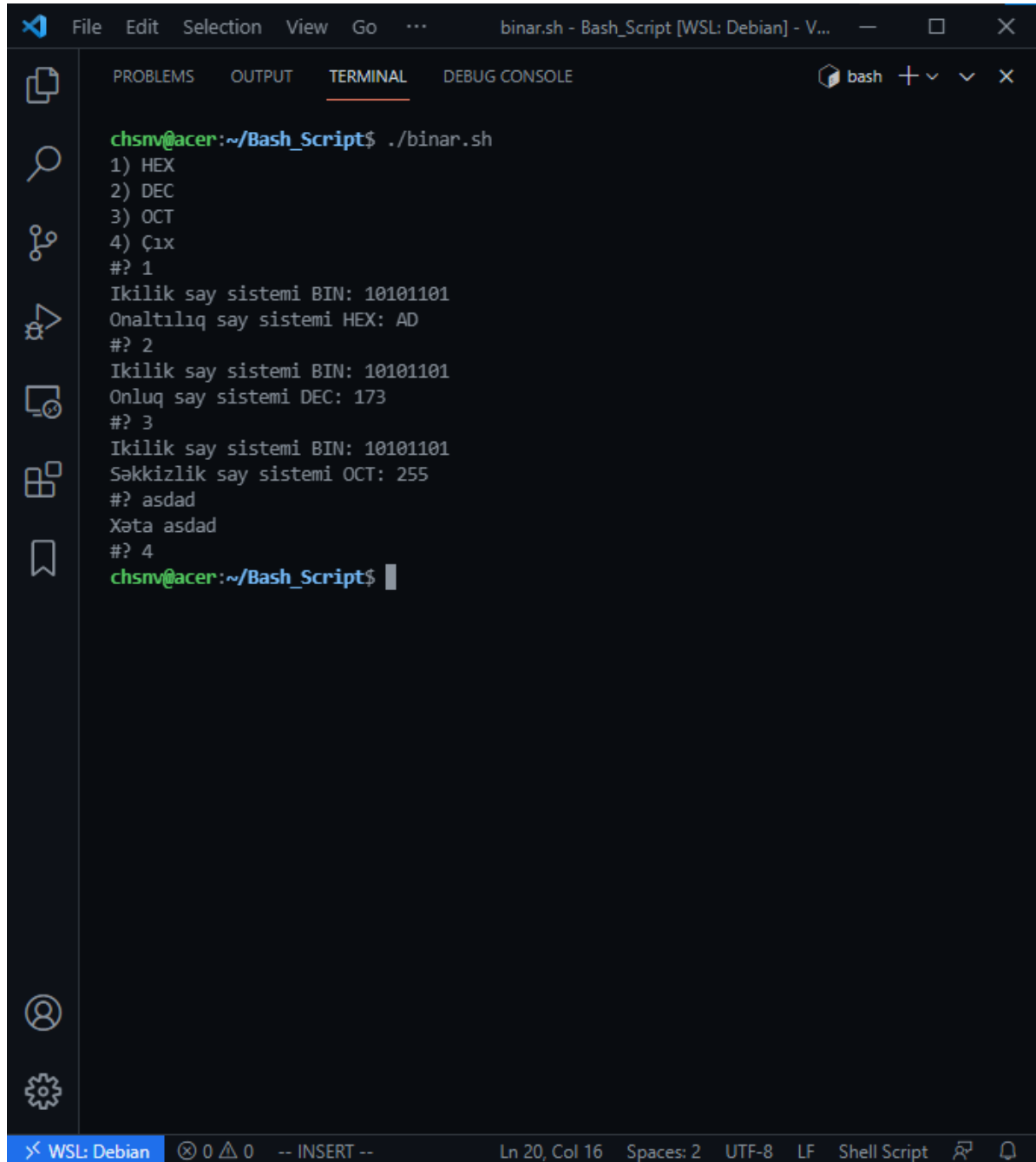
Binar **BİN** rəsəmsal sistemdən **HEX**, **DEC** və **OCT** çevrilmə funksiyaları



```
1  #!/bin/bash
2  read_bin () { read -p "İkilik say sistemi BIN: " bin;}
3  function bin_to_hex () {
4      cavab=$(bc <<< "ibase=2;obase=G;$bin")
5      echo "Onaltılıq say sistemi HEX: "$cavab
6  }
7  function bin_to_dec () {
8      cavab=$(bc <<< "ibase=2;obase=A;$bin")
9      echo "Onluq say sistemi DEC: "$cavab
10 }
11 function bin_to_oct () {
12     cavab=$(bc <<< "ibase=2;obase=8;$bin")
13     echo "Səkkizlik say sistemi OCT: "$cavab
14 }
15 select s in HEX DEC OCT Çıx; do # Menu select və case
16     case $s in
17         HEX)
18             read_bin
19             bin_to_hex;;
20         DEC)
21             read_bin
22             bin_to_dec;;
23         OCT)
24             read_bin
25             bin_to_oct;;
26         Çıx)
27             break;;
28         *)
29             echo "Xəta $REPLY";;
30     esac
31 done
32
```


Binar "Binary"

Binar **BİN** rəsəmsal sistemdən **HEX**, **DEC** və **OCT** çevrilmə nəticəsi



```
File Edit Selection View Go ... binar.sh - Bash_Script [WSL: Debian] - V...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE bash + v v x

chsnv@acer:~/Bash_Script$ ./binar.sh
1) HEX
2) DEC
3) OCT
4) Çıx
#? 1
İkilik say sistemi BIN: 10101101
Onaltılıq say sistemi HEX: AD
#? 2
İkilik say sistemi BIN: 10101101
Onluq say sistemi DEC: 173
#? 3
İkilik say sistemi BIN: 10101101
Səkkizlik say sistemi OCT: 255
#? asdad
Xəta asdad
#? 4
chsnv@acer:~/Bash_Script$
```

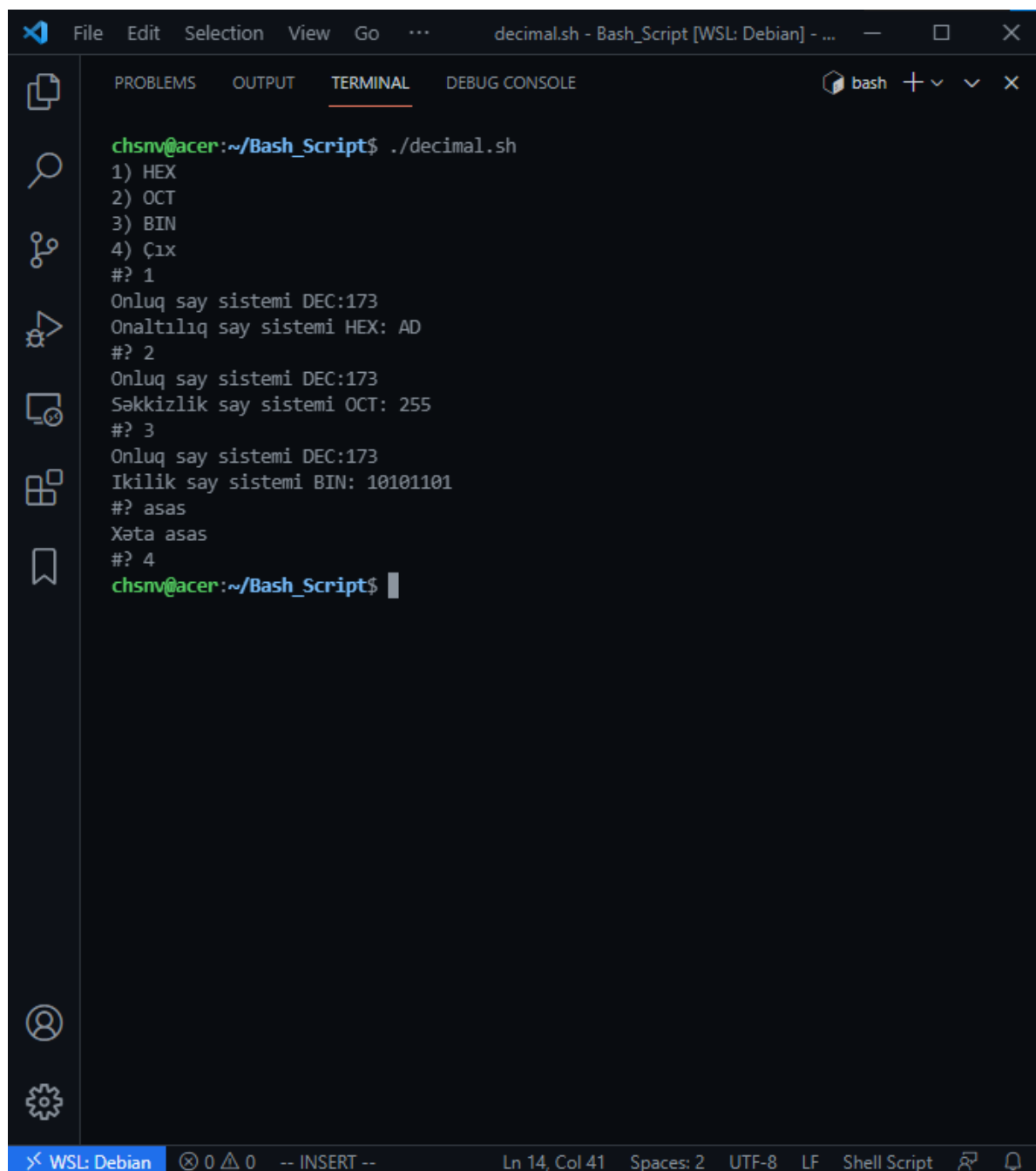
Onluq "Decimal"

Decimal DEC rəsəmsal sistemdən HEX, OCT və BIN çevrilmə funksiyaları

```
decimal.sh
1  #!/bin/bash
2  read_dec () { read -p "Onluq say sistemi DEC:" dec; }
3  function dec_to_hex () {
4      cavab=$(bc <<< "ibase=A;obase=G;$dec")
5      echo "Onaltılıq say sistemi HEX:" $cavab
6  }
7  function dec_to_oct () {
8      cavab=$(bc <<< "ibase=A;obase=8;$dec")
9      echo "Səkkizlik say sistemi OCT:" $cavab
10 }
11 function dec_to_bin () {
12     cavab=$(bc <<< "ibase=A;obase=2;$dec")
13     echo "İkilik say sistemi BIN:" $cavab
14 }
15 select s in HEX OCT BIN Çıx; do # Menu select və case
16     case $s in
17         HEX)
18             read_dec
19             dec_to_hex;;
20         OCT)
21             read_dec
22             dec_to_oct;;
23         BIN)
24             read_dec
25             dec_to_bin;;
26         Çıx)
27             break;;
28         *)
29             echo "Xəta $REPLY";;
30     esac
31 done
```

Onluq "Decimal"

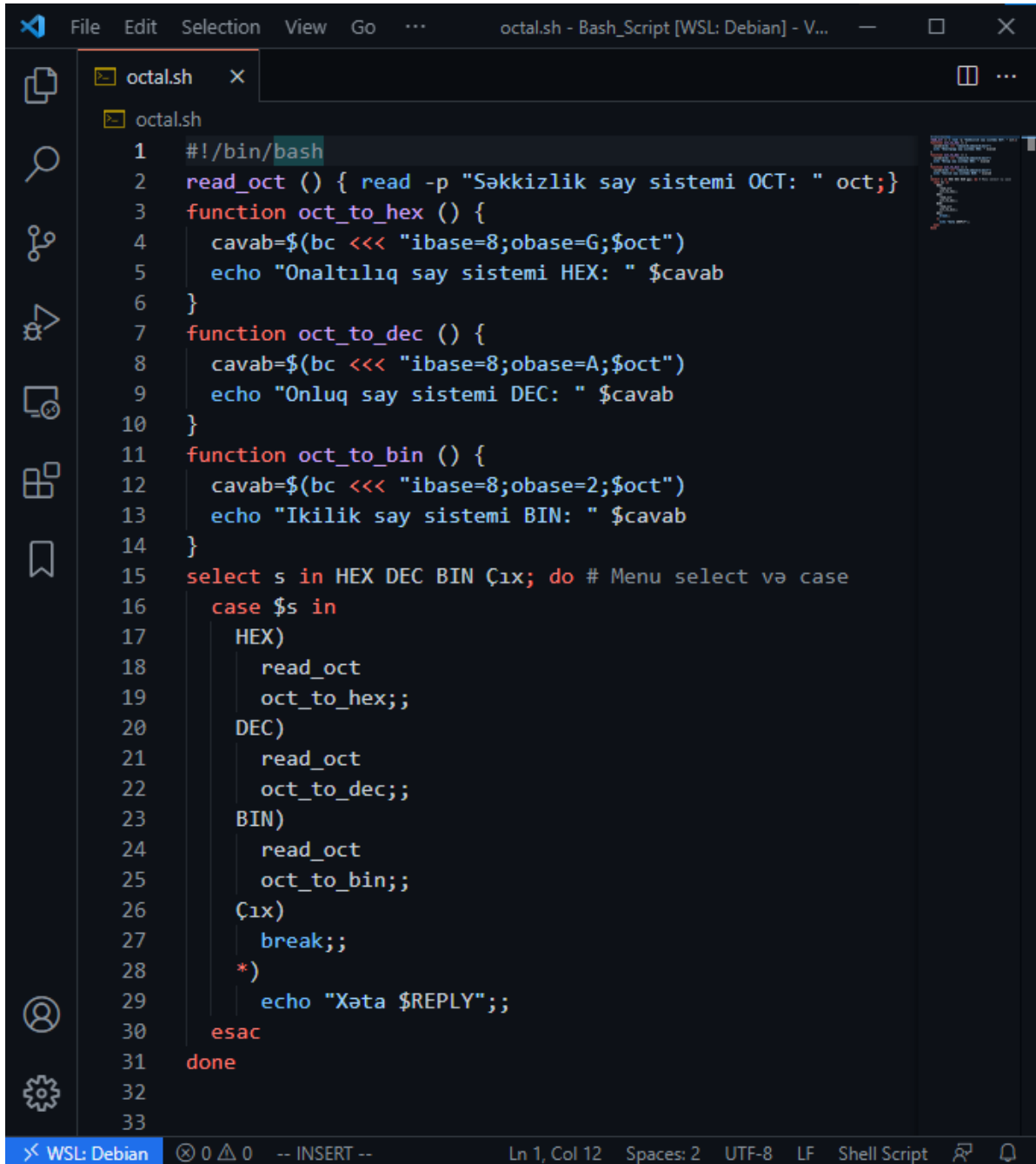
Decimal **DEC** rəsəmsal sistemdən **HEX**, **OCT** və **BİN** çevrilmə nəticəsi



```
chsnv@acer:~/Bash_Script$ ./decimal.sh
1) HEX
2) OCT
3) BIN
4) Çıx
#? 1
Onluq say sistemi DEC:173
Onaltılıq say sistemi HEX: AD
#? 2
Onluq say sistemi DEC:173
Səkkizlik say sistemi OCT: 255
#? 3
Onluq say sistemi DEC:173
İkilik say sistemi BIN: 10101101
#? اساس
Xəta اساس
#? 4
chsnv@acer:~/Bash_Script$
```

Səkkizlik "Octal"

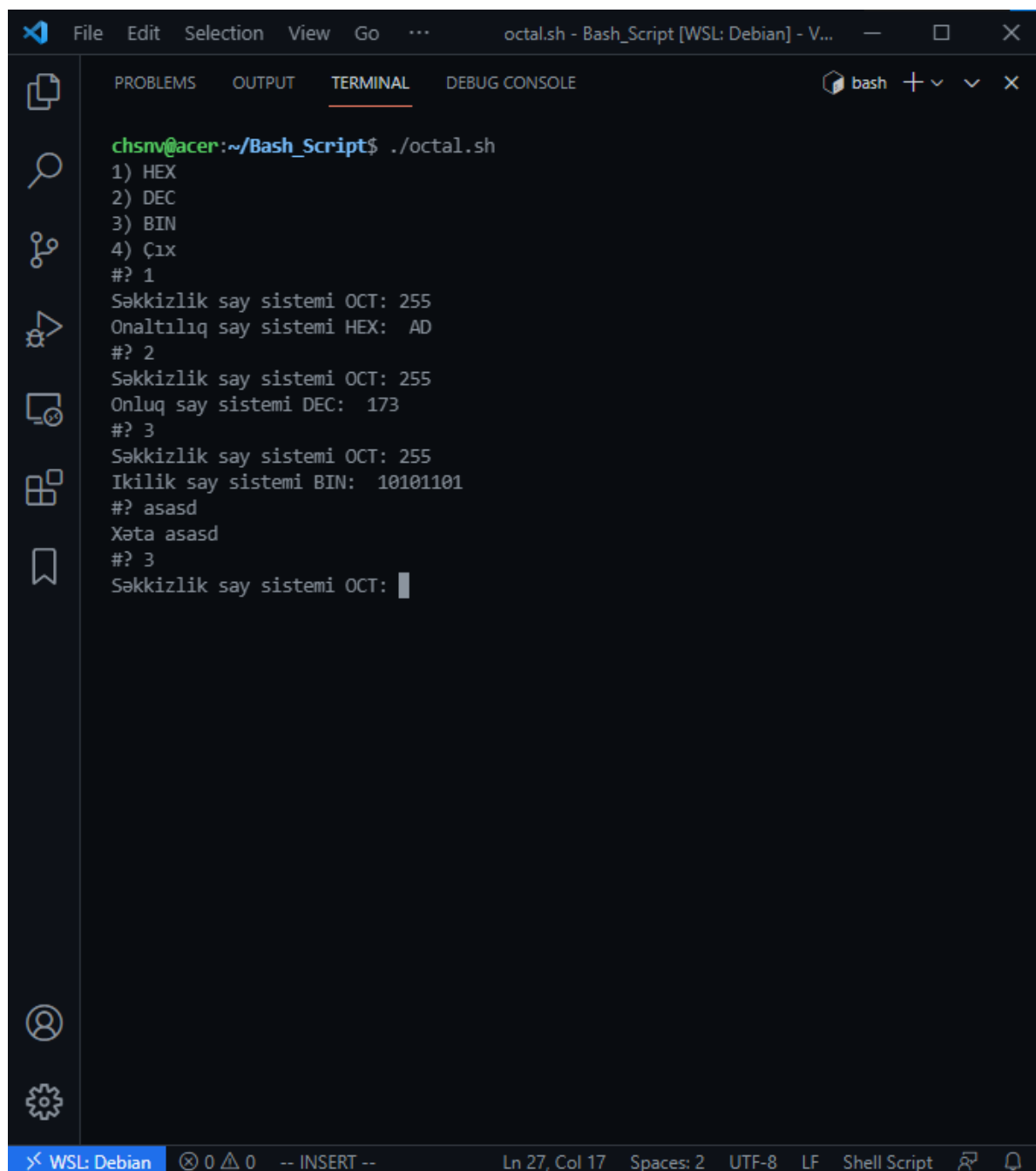
Octal **OCT** rəsəmsal sistemdən **HEX**, **DEC** və **BIN** çevrilmə funksiyaları



```
1  #!/bin/bash
2  read_oct () { read -p "Səkkizlik say sistemi OCT: " oct;}
3  function oct_to_hex () {
4      cavab=$(bc <<< "ibase=8;obase=G;$oct")
5      echo "Onaltılıq say sistemi HEX: " $cavab
6  }
7  function oct_to_dec () {
8      cavab=$(bc <<< "ibase=8;obase=A;$oct")
9      echo "Onluq say sistemi DEC: " $cavab
10 }
11 function oct_to_bin () {
12     cavab=$(bc <<< "ibase=8;obase=2;$oct")
13     echo "İkilik say sistemi BIN: " $cavab
14 }
15 select s in HEX DEC BIN Çıx; do # Menu select və case
16     case $s in
17         HEX)
18             read_oct
19             oct_to_hex;;
20         DEC)
21             read_oct
22             oct_to_dec;;
23         BIN)
24             read_oct
25             oct_to_bin;;
26         Çıx)
27             break;;
28         *)
29             echo "Xəta $REPLY";;
30     esac
31 done
32
33
```

Səkkizlik "Octal"

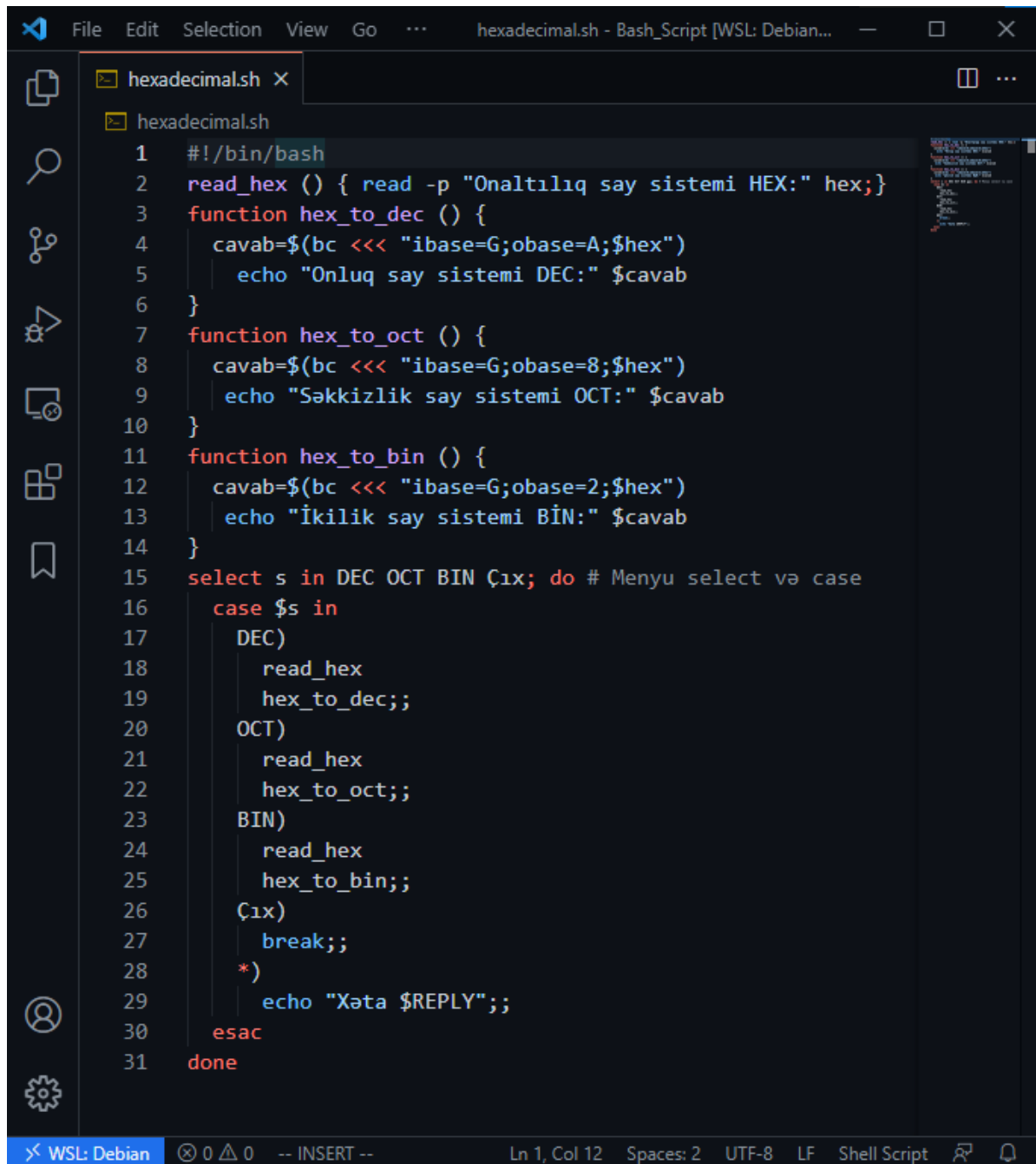
Octal **OCT** rəsəmsal sistemdən **HEX**, **DEC** və **BIN** çevrilmə nəticəsi



```
chsmv@acer:~/Bash_Script$ ./octal.sh
1) HEX
2) DEC
3) BIN
4) Çıx
#? 1
Səkkizlik say sistemi OCT: 255
Onaltılıq say sistemi HEX: AD
#? 2
Səkkizlik say sistemi OCT: 255
Onluq say sistemi DEC: 173
#? 3
Səkkizlik say sistemi OCT: 255
İkilik say sistemi BIN: 10101101
#? asasd
Xəta asasd
#? 3
Səkkizlik say sistemi OCT: █
```

Onaltılıq "Hexadecimal"

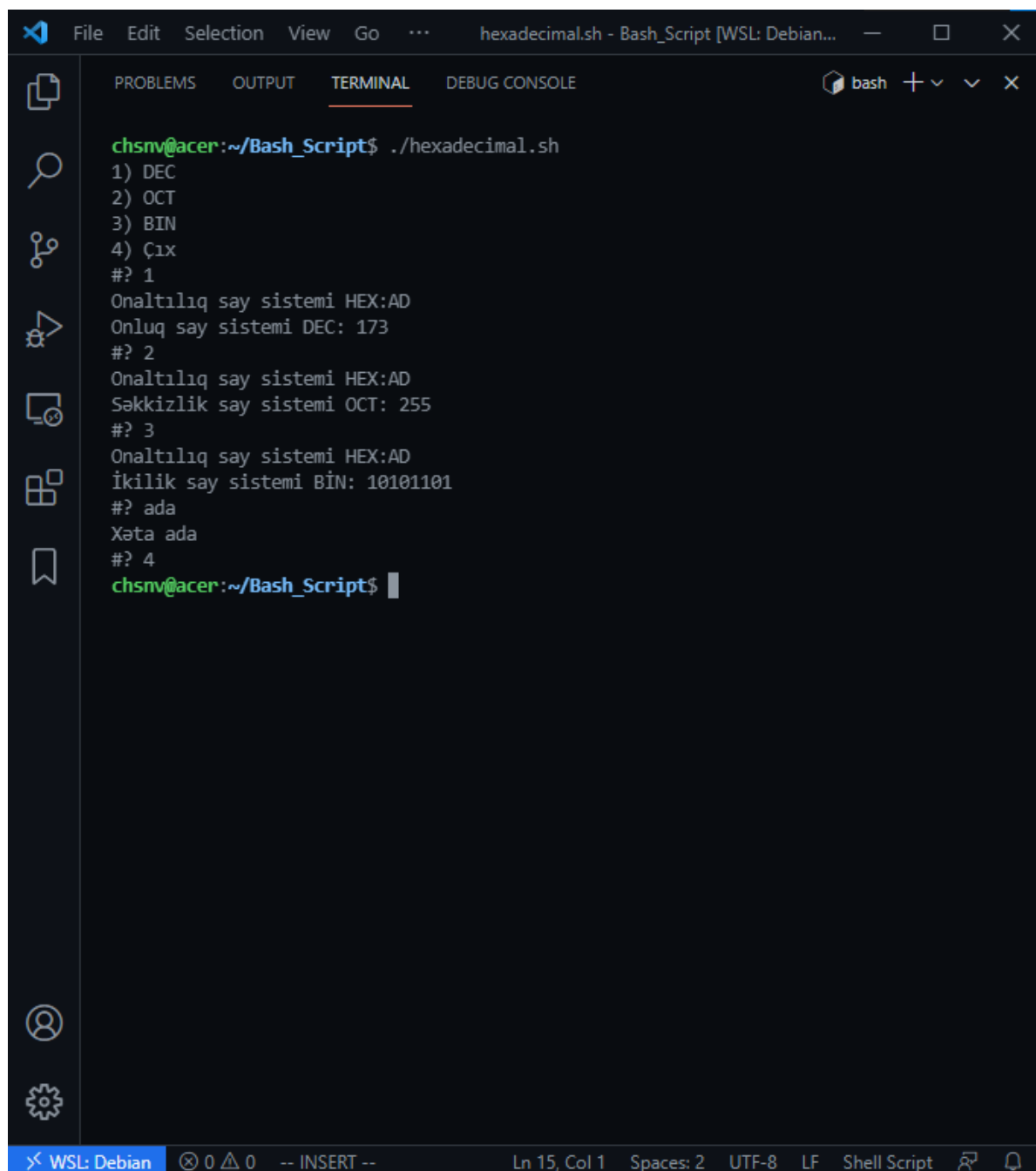
Hexadecimal **HEX** rəsəmsal sistemdən **DEC**, **OCT** və **BİN** çevrilmə funksiyaları



```
hexadecimal.sh
1  #!/bin/bash
2  read_hex () { read -p "Onaltılıq say sistemi HEX:" hex;}
3  function hex_to_dec () {
4      cavab=$(bc <<< "ibase=G;obase=A;$hex")
5      echo "Onluq say sistemi DEC:" $cavab
6  }
7  function hex_to_oct () {
8      cavab=$(bc <<< "ibase=G;obase=8;$hex")
9      echo "Səkkizlik say sistemi OCT:" $cavab
10 }
11 function hex_to_bin () {
12     cavab=$(bc <<< "ibase=G;obase=2;$hex")
13     echo "İkilik say sistemi BİN:" $cavab
14 }
15 select s in DEC OCT BİN Çıx; do # Menu select və case
16     case $s in
17         DEC)
18             read_hex
19             hex_to_dec;;
20         OCT)
21             read_hex
22             hex_to_oct;;
23         BİN)
24             read_hex
25             hex_to_bin;;
26         Çıx)
27             break;;
28         *)
29             echo "Xəta $REPLY";;
30     esac
31 done
```

Onaltılıq "Hexadecimal"

Hexadecimal **HEX** rəsəmsal sistemdən **DEC**, **OCT** və **BİN** çevrilmə nəticəsi



```
File Edit Selection View Go ... hexadecimal.sh - Bash_Script [WSL: Debian...]
```

```
chsnv@acer:~/Bash_Script$ ./hexadecimal.sh
1) DEC
2) OCT
3) BIN
4) Çıx
#? 1
Onaltılıq say sistemi HEX:AD
Onluq say sistemi DEC: 173
#? 2
Onaltılıq say sistemi HEX:AD
Səkkizlik say sistemi OCT: 255
#? 3
Onaltılıq say sistemi HEX:AD
İkilik say sistemi BİN: 10101101
#? ada
Xəta ada
#? 4
chsnv@acer:~/Bash_Script$
```

WSL: Debian 0 0 -- INSERT -- Ln 15, Col 1 Spaces: 2 UTF-8 LF Shell Script