

1. Feladat: Készítse el masol.sh scriptet, mely bekéri a forrás és a cél nevét, majd a cp parancsnak megfelelően elvégzi a másolást!

```
klettner@panda:~/gy5$ masol.sh
Forrás: szam.txt
Cél:    szam.dat
klettner@panda:~/gy5$ ls szam.*
szam.dat szam.txt
klettner@panda:~/gy5$
```

2. Feladat: Írjunk szokeres.sh scriptet, aminek segítségével keressük meg azokat a fájlokat egy paraméterként átadott könyvtárban és alkönyvtáraiban, amikben egy szintén paraméterként átadott szövegrészlet megtalálható.

```
klettner@panda:~/gy5$ szokeres.sh ~ csap
/afs/inf.elte.hu/user/k/kl/klettner/gy4/felhasznalok.txt
/afs/inf.elte.hu/user/k/kl/klettner/felhasznalok.txt
klettner@panda:~/gy5$
```

3. Feladat: Írjunk nagybetu.sh scriptet, mely kicseréli egy beolvasott szöveg betűit csupa nagybetűre! Ékezetes betűkre is működjön!

```
klettner@panda:~/gy5$ nagybetu.sh
Kérem a szöveget: Hátamon a púpom, megbékélnék vele ...
Nagybetűkkel:    HÁTAMON A PÚPOM, MEGBÉKÉLNÉK VELE ...
klettner@panda:~/gy5$
```

4. Feladat: Írjunk hanyjegyu.sh scriptet, mely bekér egy számot, majd kiírja, hogy a szám egyjegyű, kétjegyű, háromjegyű, vagy háromnál több jegyű! Negatív számokra is működjön!

```
klettner@panda:~/gy5$ hanyjegyu.sh
Kérek egy számot: -54
A szám kétjegyű!
klettner@panda:~/gy5$ hanyjegyu.sh
Kérek egy számot: 1478
A szám több mint háromjegyű!
klettner@panda:~/gy5$
```

5. Feladat: Írjunk napszak.sh scriptet, kiírja, hogy reggel (5-8 óra), délelőtt (9-12 óra), délután (13-19 óra), vagy éjszaka (20-4 óra) van. A pontos időt a rendszeridőből kérdezze le!

```
klettner@panda:~/gy5$ napszak.sh
Délután van.
klettner@panda:~/gy5$
```

6. Feladat: Módosítsuk a .profile állományt úgy, hogy a napszaknak megfelelően köszöntsön!

```
Volume Name          Quota      Used %Used  Partition
user.klettner         512000     32783     6%       15%

Kellemes délutánt kedves Péter!
Jó munkát!

klettner@valerie:~$
```

7. Feladat: Írjunk until_menu.sh scriptet, mely egy, az alábbi minta szerinti választásos menüt valósít meg! A választás után 2 másodpercig jelenítse meg a megfelelő szöveget! A képernyőt a clear paranccsal lehet törölni.

<pre>Válasszon! 1: Első menüpont 2: Második menüpont 3: Kilépés Adja meg a megfelelő számot: 5 Rossz opciót adott meg! </pre>	<pre>Válasszon! 1: Első menüpont 2: Második menüpont 3: Kilépés Adja meg a megfelelő számot: 2 A másodikat választotta! </pre>
--	---

8. Feladat: Írjunk fakt.sh scriptet, ami kiszámolja egy paraméterben kapott szám faktoriálisát! Ellenőrizzük, hogy pontosan egy paramétert kaptunk-e! Azt is ellenőrizzük, hogy a paraméter nem negatív és nem szöveg!

```
klettner@panda:~/gy6$ faktor.sh -5
Kérem ne adjon negatív paramétert!
klettner@panda:~/gy6$ faktor.sh d
Kérem csak számjegyeket írjon paraméterként!
klettner@panda:~/gy6$ faktor.sh 5d
Kérem csak számjegyeket írjon paraméterként!
klettner@panda:~/gy6$ faktor.sh 5 6
Kérem pontosan egy paramétert adjon!
klettner@panda:~/gy6$ faktor.sh 0
0 faktoriálisa: 1
klettner@panda:~/gy6$ faktor.sh 5
5 faktoriálisa: 120
klettner@panda:~/gy6$
```

9. Feladat: Írjunk sorba_vege_ig.sh scriptet, ami addig olvas be sorokat, míg a „vége” szót be nem írjuk, majd írja ki a beírt sorokat ABC sorrendbe rendezve! A vége szó ne szerepeljen a kiírásban és olyan üres sorok se, melyeket nem adtunk meg! Használjunk ideiglenes fájlt!

```
klettner@panda:~/gy6$ sorba_vege_ig.sh
körte
alma
3
szilva
vége
3
alma
körte
szilva
klettner@panda:~/gy6$
```

10. Feladat: Készítse el osszead_param.sh scriptet, amelyik a paraméterként kapott számokat összeadja és az összeget a szabványos kimenetre írja! A paraméterként megadott számok helyett megadhatjuk a --help kapcsolót, aminek hatására írjon ki rövid ismertetőt a programról!
11. Feladat: Készítse el felhasznalok.sh scriptet, mely a getent passwd parancs segítségével megjeleníthető listát két irányba görgethetően jeleníti meg a képernyőn.

12. Feladat: Készítse el `szamol.sh` scriptet, mely a paraméterként kapott két számot összeadja, kivonja, összeszorozza, és elosztja egymással.

```
klettner@valerie:~/gy6$ ./szamol.sh 50 15
50+15=65
50-15=35
50*15=750
50/15=3
klettner@valerie:~/gy6$
```

13. Feladat: Készítse el `szamol2.sh` scriptet, mely bekér két számot, majd az *osszeg* változóba helyezi a bekért számok összegét, a *kulonbseg* változóba helyezi a bekért számok különbségét, a *szorzat* változóba helyezi a bekért számok szorzatát, és a *hanyados* változóba helyezi a bekért számok hányadosát. Ezután a létrehozott négy változót környezeti változókká alakítja. Végül kilistázza a környezeti változókat a képernyőre két irányba görgethetően.
14. Feladat: Készítse el `szamol3.sh` scriptet, mely a 15. feladatban leírtakat hajtja végre azzal a különbséggel, hogy ha kettőnél kevesebb paramétert adunk meg, akkor a hiányzó paraméterek értékét bekéri a felhasználótól.

```
klettner@valerie:~/gy6$ szamol3.sh 50 15
50+15=65
50-15=35
50*15=750
50/15=3
klettner@valerie:~/gy6$ szamol3.sh 50
Kérem a második számot: 15
50+15=65
50-15=35
50*15=750
50/15=3
klettner@valerie:~/gy6$ szamol3.sh
Kérem az első számot: 50
Kérem a második számot: 15
50+15=65
50-15=35
50*15=750
50/15=3
klettner@valerie:~/gy6$
```

15. Feladat: Hozzuk létre sz.txt fájlt az alábbi tartalommal:

orrszarvú

traktor

orv

oszlopsor

kor

kortalan

orátor

Ezután írassuk ki sz.txt összes *or* betűkapcsolatot tartalmazó sorát!

16. Feladat: Listázzuk ki a 15. feladatban létrehozott sz.txt fájl összes *or*-ral kezdődő sorát!

17. Feladat: Listázzuk ki a 15. feladatban létrehozott sz.txt fájl összes *or*-ral végződő sorát!

18. Feladat: Hozzuk létre or.sh scriptet, mely létrehozza or.txt fájlt, amibe a 15., 16. és 17. feladat eredményeit helyezi el a mintának megfelelő formátumban. Ha egymás után többször futtatjuk a scriptet, akkor is ugyanez legyen az or.txt tartalma, ne ismétlődjön!

```
klettner@valerie:~/gy5$ or.sh
klettner@valerie:~/gy5$ cat or.txt
Tartalmazza or-t:
orrszarvú
traktor
orv
oszlopsor
kor
kortalan
orátor

or-ral kezdődnek:
orrszarvú
orv
orátor

or-ral végződnek:
traktor
oszlopsor
kor
orátor
klettner@valerie:~/gy5$
```