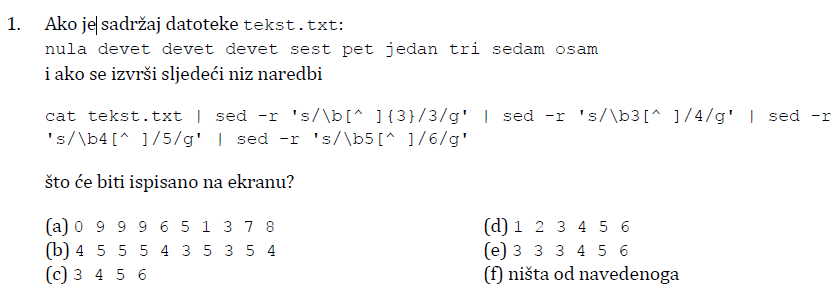
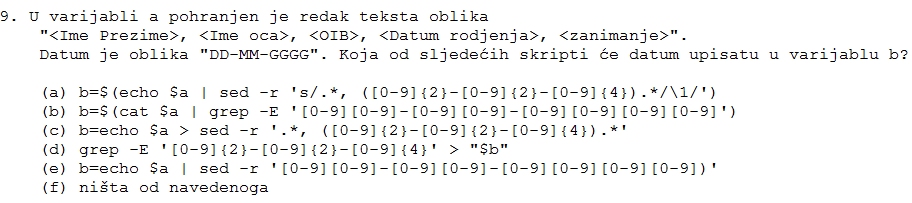
**

Prvi sed traži sljedeće: graničnik, 3 puta nePraznina - i to mijenja sa brojem 3  
nula -> nul.a -> 3a  
devet -> dev.et -> 3et (x3)  
sest -> ses.t -> 3t  
pet -> pet. -> 3  
jedan -> jed.an -> 3an  
tri -> tri. -> 3  
sedam -> sed.am -> 3am  
osam -> osa.m -> 3m  
  
Drugi sed traži sljedeće: graničik, broj 3, 1 puta nePraznina - i to mijenja sa brojem 4  
3a -> 3a. -> 4  
3et -> 3e.t -> 4t (x3)  
3t -> 3t. -> 4  
3 -> 3 -> 3 (ostaje trojka jer traženog niza)  
3an -> 3a.n -> 4n  
3 -> 3 -> 3  
3am -> 3a.m -> 4m  
3m -> 3m. -> 4  
  
Treći sed traži sljedeće: graničkih, broj 4, 1 puta nePraznina - i to mijenja sa brojem 5  
4 -> 4 -> 4  
4t -> 4t. -> 5 (x3)  
4 -> 4 -> 4  
3 -> 3 -> 3  
4n -> 4n. -> 5  
3 -> 3 -> 3  
4m -> 4m. -> 5  
4 -> 4 -> 4  
  
Četvrti sed pak traži: graničnik, broj 5, 1 puta nePraznina - i to mijenja sa brojem 6  
  
Međutim mi takvu situaciju više nemamo stoga ono što nam je pretposljednjim sedom ostalo je konačno rješenje:  
  
4 5 5 5 4 3 5 3 5 4  
  
Još kad bi mi se to dalo uređivati meh  
  
EDIT: cat tekst.txt naravno pajpa ispis u sedove, nadam se da se to podrazumijeva

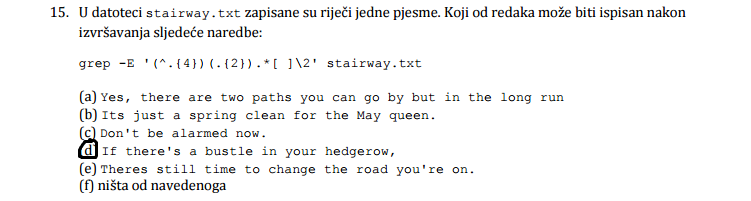
2) 

\1 ti je rezultat prve zagrade, znaci **.\*, ([0-9]{2}-[0-9]{2}-[0-9]){4}) .\*** ce se zamjeniti sa **([0-9]{2}-[0-9]{2}-[0-9]{4})** (ali ne bilo kojom kombinacijom brojeva nego **bas tom** koju je pronasao u tom redu

1 ti predstavlja da se koristei izraz u zagradama ovaj ([][][]), tipa da imas vise izraza u zagradama tipa 's/(prvi)+(drugi)/\1\.\2', 1 ti je izraz prvi, 2 ti je izraz drugi  
  
u ovom zadatkuide tak, ispises varijablu $a, i onda trazis izraz koji pase .\*, (datum).\* -> tako da kad ce trazit nasel bude "OIB, datum, zanimanje da mu pase taj izraz, a u zagrade si si stavil samo datum, i onda zamjenis sve to sa izrazom 1 (1=datum)  
  
s - oznacava da zelis nekaj mjenjat  
's/uzorak/zamjena/'

*8. Što će se ispisati na ekranu kada se izvrši sljedeći niz naredbi?  
uvjet="utorak je pravi dan za ispit"  
case $uvjet in  
utorak ) echo "prvi";;  
.\*ispit ) echo "drugi";;  
\*pravi\* ) echo "treci";;  
[^uto]uto\* ) echo "cetvrti";;  
\*dan ) echo "peti";;  
esac  
  
  
zar ne bi ovo trebalo ispisati "drugi", a ne "treci"?*

U caseu se koristi globbing pa taj .\*ispit zapravo traži izraz "točka + proizvoljan niz znakova + ispit". Ulogu točke u regexu bi tu obavljao upitnik.  
  
Nisam ni ja znao, sad sam pogledao

*[](http://postimage.org/)*

prva zagrada ti uzima bilo koja 4 znaka, druga bilo koja 2  
  
znači imaš:  
1 = If t  
2 = he  
  
onda uzimaš znakove sve dok ti se ne pojavi prazan znak iza kojeg slijedi broj 2, odnosno he  
  
ako to primjeniš na sve vidiš da samo d) ima tako

ovaj (^.{4}) znaci 4 proizvoljna znaka s pocetka, (.{2}) znaci sljedeca 2 proizvoljna znaka, .\* znaci sljedecih 0 ili n proizvoljnih znakova, nakon tih znakova mora slijedit praznina [ ] i \2 (sto znaci znaci da nakon tih proizvoljnih znakova i razmaka slijede znakovi koji odgovaraju ovom drugom skupu znakova odrednih s (.{2}) ),   
Sad ak pogledas niz prva 4 znaka su "If t", sljedeca 2 znaka su "he", sad ce ici po svim znakovima zbog .\* dok ne naide na nesto sto odgovara " he"(ovaj razmak unutra se odnosi na [ ]). U ostalim nizovima vidis da nakon sto naide na takva dva znaka ni jedan u daljenjem tekstu ne odgovara njima

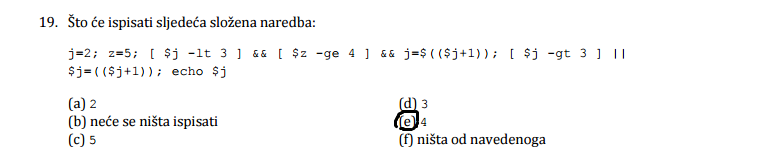
zato jer ovaj pattern matcha  
\1 = pocetak reda, 4 znaka, bilo koja  
\2 = bilo koja 2 znaka  
bilo sto bilo koliko puta  
nakon toga mora naci jedan razmak i \2  
za d):  
\1 = If t  
\2 = he  
i sad pogledaj hedgerow

(^.{4}) -> prva četiri znaka na pocetku reda i zbog zagrada ta matchana 4 znaka kasnije u regexu mozes referencirat s \1  
  
(.{2}) -> 5. i 6. znak u redu, zbog zagrada kasnije mozes referencirat s \2  
  
.\*[ ]\2 -> u principu "jel postoji riječ koja zapocinje s 5. i 6. znakom koji sam maloprije matchao?"  
  
a)  
prva četiri znaka "Yes,"  
sljedeca dva " t"  
nijedna od sljedecih rijeci ne zapocinje s " t"  
  
b)  
prva cetiri "Its "  
sljedeca dva "ju"  
nijedna od sljedecih rijeci na zapocinje s "ju"  
  
c)  
prva cetiri "Don'"  
sljedaca dva " b"  
opet nista...  
  
d)  
prva cetiri "if t"  
sljedeca dva "he"  
zadnja rijec "hedgerow" zapocinje s "he" -> MATCH

prepisi u normalan format a ne u jednom redu

|  |
| --- |
| **Source kod:** |
| for i in 6 3 5  do  j="$i+2"  k="$i+$j"  echo -n $k  done > d  echo $(($(cat d))) |
|  |

vjerojatno ti pokuplja ovaj ; na for i in 6 3 5; pa onda u d ima gluposti  
  
ovo gore izbacuje 70, izgenerira taj cijeli izraz i onda se to izracuna u $((...))

*[](http://postimage.org/)*

Kad imaš && sljedeći izraz će se izvršiti ako prethodni vrati 0 što označava da je uspješno izvršen.  
  
j je manje od 3, pa ideš na sljedeći, g je veće ili jednako 4 pa ideš na sljedeći, postavljaš j na 4.  
  
Kod || se sljedeći izraz izvrši ako prethodni nije uspio. Budući da je j veće od 3, onda preskačeš ovaj sljedeći koji bi uvećao j na 5. Kad j ne bi bio veći od 3, taj bi vratio 1 (false) i izvršila bi se naredba iza ||.  
  
**Nisam rekao dobro.** Mislio sam da je j na početku 3. Onda kod ovog || j nije veće od 3 (jer je 3) pa će se izvršiti i sljedeća koja ga povećava na 4. Valjda.  
I da, kao što miwako kaže fali $. Ne znam jeli namjerno.

*11. U datoteci pjesma.txt zapisane su riječi jedne pjesme. Koji od redaka može biti ispisan nakon izvođenja sljedeće naredbe:   
grep -E -i '\b([a-zA-Z]{3})\b.\*\b\1\b' pjesma.txt  
  
(a) in bending o'er you, queen of every charm,  
(b) The suns were beauteous in those twilights warm.  
\* (c) and space profound, and strong life's pulsing flood,  
(d) I thought I breathed the perfume in your blood.  
(e) Lovely the suns were in those twilights warm,  
(f) ništa od navedenoga*

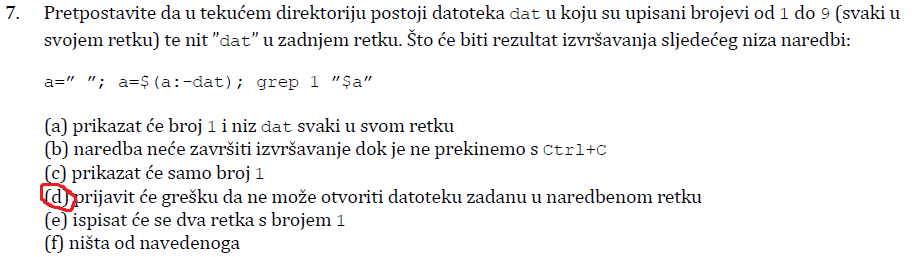
*može pomoć oko ovog? ustvari dali može potvrda mog razmišljanja http://www.fer2.net/images/smilies/help.gif  
  
znači ovak 1. dio je ovaj ([a-zA-Z]{3})  
2. dio je ovaj .\*  
ovo \1 nam kaže da uzmemo prvi dio, koji odabere prva tri znaka koji su između a-z ili A-Z što je u c.) slučaju "and" i onda gleda dali se taj "and" pojavljuje još negdje u ostatku teksta.  
  
Slično kao i zadatak 15 koji je tu već diskutiran. Jel ovo dobro razmišljanje?*

pa.. mislim da te malo bune neke stvari tak da bih nadodala:  
  
\1 -> ovo 1 se misli na 1. **zagradu**, tako da po toj logici \2 ne postoji.   
  
takodjer bih spomenula i ove \b, sto oznacavaju granicu -> u ovom slucaju mi trazimo rijec koja se sastoji od tri alpha znaka, koja ce se takodjer i kasnije javiti unutar linije nakon nekog smeca .\* (zato ide taj \1)  
  
ja sam to tak shvatila http://www.fer2.net/images/smilies/biggrin.gif  
  
tak da nisi nista previse fulao, al eto moj komentar na zadatak http://www.fer2.net/images/smilies/biggrin.gif

Izvorni post od **ijsfd** [Pogledaj post](http://www.fer2.net/showthread.php?p=2399222#post2399222)

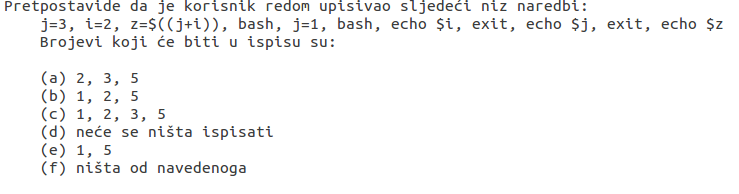
*4. Koji su prvih pet znakova koje će ispisati sljedeća naredba:  
for i in [1..5]; do j=0; while [ $j -lt $i ]; do echo $j; j=$(($j+1)); done; done  
(a) 4 3 2 1 0 (d) 0 0 1 0 1  
(b) 0 1 2 3 4 (e) neće se ništa ispisati  
(c) 0 0 1 2 3 (f) ništa od navedenoga  
  
u rješenjima piše da je točno rješenje d, ja bih rekao da je c, jel me može netko ispraviti?*

1. krug i = 1, piše se 0  
   2. krug i = 2, piše se 0, 1  
   3. krug i = 3, piše se 0, 1, 2  
     
   prvih 5 je 0 0 1 0 1

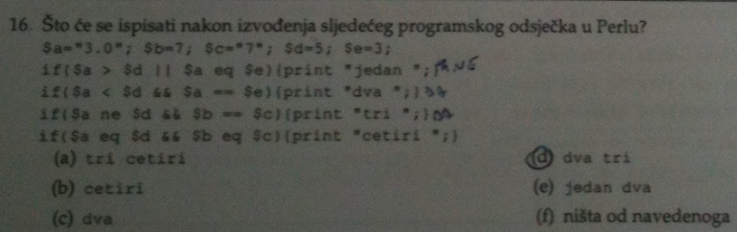
**

:- koristi postavljenu vrijednost ako postoji, a ona je " " u ovom zadatku, što nije datoteka i ne može se otvoriti

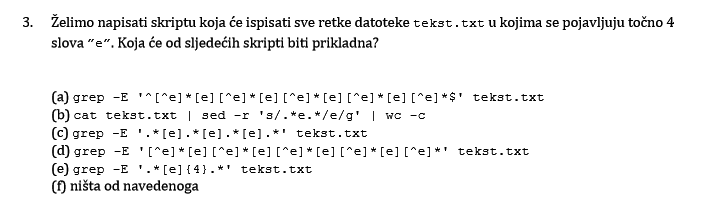
1. a=" " --> postavi vrijednost varijable a na razmak  
   2. a=${a:-dat} ---> ovaj izraz s desne strane znači širenje varijable a na način ako varijabla a nije postavljena ili je prazna onda je postavi na vrijednost dat samo u ovom koraku ( zbog ovog :- ). U suprotnom je postavi na vrijednost a. Budući je naša varijabla a pod jedan postavljena, onda ovaj korak samo kaže a=" "  
   3. grep 1 "$a" - dohvati liniju u kojoj piše 1 iz datoteke "$a"-->širenje imena varijable, dakle na tom mjestu piše ono što piše u varijabli a a to je razmak.  
   kako mi nemamo datoteku čije je ime " ", javlja nam grešku.

**

Kada upišemo bash onda otvaramo novu ljusku u kojoj se nalaze samo one varijable koje su varijable okoline (environment).  
Npr. ako upišemo j=3, to je obična varijabla koja je vidljiva samo u ljusci u kojoj se trenutno nalazimo, što znači ako upišemo bash i uđemo u novu ljusku više ne vidimo tu varijablu. Međutim ako upišem export j=3 onda ta varijabla postaje varijabla okoline što znači da će biti vidljiva u svim ljuskama.  
  
Pa krenimo na konkretan primjer:  
j=3; i=2; z=5; #sve su nam to obične varijable, dakle vidljive u trenutnoj ljusci na koju si pijavljen  
bash # otvaramo novu podljsku, nazovimo je ljuska 2, gdje još nismo postavili varijable  
j=1 # ova varijabla je vidljiva u ljusci 2  
bash #ulazimo u novu ljusku, nazovimo je ljuska 3  
echo $i #ne ispisuje nam se ništa jer nismo postavili ovu varijablu u ljusci 2  
exit #izlazimo iz ljuske 3 i vraćamo se u ljusku 2  
echo $j # u ljusi 2 smo postavili vrijednost ove varijable pa se ispisuje 1  
exit #izazimo iz ljuske 2  
echo $z #nalazimo se u ljusci 1 i ispisuje se 5  
  
Izlaz: 1 5

**

Da, ovisno o tome koju operaciju radiš, castaju ti se stringovi u broj ili obratno. S time da ti je 3.0 isto što i 3 (EDIT: samo u broju, u stringu su "3.0" i "3" različiti naravno, maj bed)

[](http://www.deviantpics.com/images/2016/05/04/Screenshot_3.png)

jer ako ti grep matcha d) isprintat će ga, ali nakon toga u istoj liniji možeš opet imati e  
  
npr. "dededededddde" prolazi u d)

pa tocan je A, jer d) ce ti matchat prvi dio u kojem se pojavljuju 4 puta e, i nakon toga se moze pojaviti jos e znakova

1)Koja od sljedćih naredbi će u Perlu ispravno sortirani listu brojeva @brojevi:  
  
a)sort @brojevi  
b)@brojevi=sort -n brojevi  
c)@brojevi=sort @brojevi  
\*d)**@brojevi=sort {$a <=> $b} @brojevi**  
e)@brojevi=sort {$a cmp $b} @brojevi  
f)ništa od navedenoga

|  |
| --- |
| *d)****@brojevi=sort {$a <=> $b} @brojevi*** *zašto je ovo točno rješenje prvog zadatka? znam da sort u perlu inace sortira po ascii kodu, ali kako nam {$a <=> $b} utjece na to?* |

Ako staviš <=> onda se lista brojeva sortira po vrijednosti. Bez toga će ti recimo 10 biti prije 9 jer ima manju ASCII vrijednost

|  |
| --- |
| Izvorni post od **Jureknez** [Pogledaj post](http://www.fer2.net/showthread.php?p=2399513#post2399513)  *Što će se ispisati izvršavanjem echo bla bla bla bla blabla| sed -r "s/(\w\w\w)\W\1/\1" šta je ova jedinica nakon \W?? zapravo, moze li neko objasnit?* |

\w oznacava slova i brojeve [A-Za-z0-9]. \W označava sve ostalo.  
  
\w\w\w znači niz od tri znaka koja su slovo ili broj. To će pronaći "bla"  
  
(\w\w\w)\W znači da na to dodaš još neki znak koji nije slovo ili broj, u našem slučaju razmak. Ovo će pronaći "bla "  
  
\1 znači da ponoviš ono što je bilo u zagradama. Vidimo da je to bilo (\w\w\w) što nam je pronašlo "bla" pa se to ponavlja i sad imamo "bla bla".  
  
Da si poslije imao još neki dio u zagradama to bi bilo \2  
  
Npr (\w\w\w)\W(\w\w) - (\w\w\w) - \1, a (\w\w) - \2.  
  
\broj zamjenjuje točno onaj izraz koji si pronašao u njegovoj odgovarajućoj zagradi.  
  
I sad to sve zamjeni sa \1 što smo već rekli da je "bla".  
  
Pa na kraju imaš:  
"bla bla" -> "bla" I to na kraju u čitavom izrazu smanji broj blaova za jedan. Budući da sed na kraju nema /g zamjena se obavlja samo na prvoj pojavi pronađenog izraza.

2)Koja naredba će ispisati ekstenzije svih datoteka u tekucem i svim poddirektorijima:  
**find . -type f -name '\*.\*'|sed -e 's/.\*\.//' | sort -u**  
  
3)Koji ispis bi se mogao dobiti primjenom naredbe echo ???[^d]\*.\*[o]  
**ld-2.5.so...**  
  
4)Kako u perlu inicijalizirat praznu listu?  
**@a=();**  
  
5)Koja će od sljedećih naredbi pokretanja skripte envar\_manipulation.sh osigurati  
da promjene nad varijablama okoline ostanu nakon izvršavanja skripte?  
**source envar\_manipulation.sh**

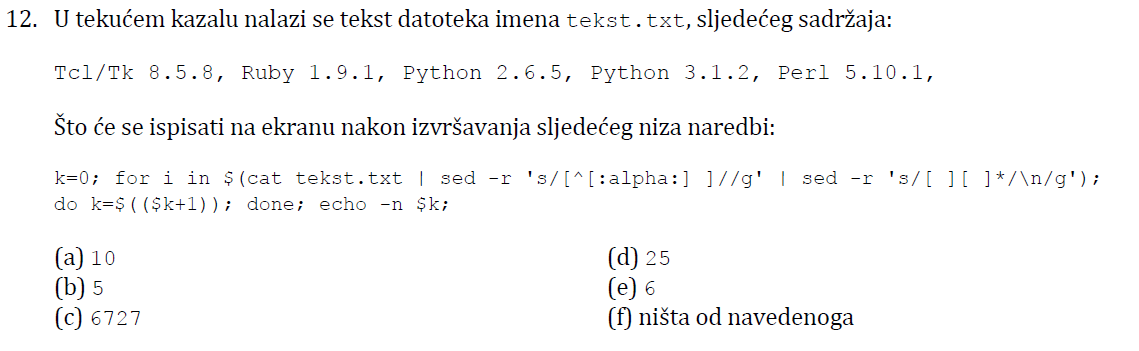
7)Izvršavaju se redom naredbe j=4,i=2,z=$((j+i)),bash,j=1,bash,echo $i,exit,echo $j,exit,echo $z.  
Što će se ispisati?  
**1 6**  
  
8)Šta će se ispisati u perlu nakon izvršavanja: @a=(1,2,3,4);@b=(5,@a,6,7);$c=@b;print $c . " : " . $b[1];  
**7 : 1**  
  
9)Izvršavaju se u ljusci naredbe: a="c";c="a";b="b";echo a>dat;echo b>dat;echo c>dat; Koji je sadržaj datoteke dat?  
**jedan redak sa znakom c**  
  
13)Kako standardni i dijagnostički ispis programa prog preusmjeriti u log.txt uz očuvanje prethodnog sadrzaja datoteke?  
a)prog >> log.txt  
**b)prog &>>log.txt**  
c)  
d)prog 1&2 >> log.txt  
e) prog 2>log.txt  
  
  
14)Tražimo datoteku unutar ~ kazala za koju znamo da sadrži deseteroznamenkasti broj koji počinje s 0036 a završava s 456,  
koja naredba nam u tome može pomoći?  
  
**grep -rE '0036[0-9]{3}456'**  
  
15)Perl, što će se ispisati: @a=/fred barney betty/; @b=qw {nesto dino}; print @a+@b;  
**5**  
  
19)Što će se ispisati na ekranu izvođenjem naredbe:  
echo $((for i in 5 7 4;do;j="$i+2";k="$i+$j";echo -n $k;done));  
**74**  
  
20)Izbrojati sve c datoteke u tekućem kazalu i podkazalima:  
**find . -regex '..\*\.c$'|wc -l**  
  
23)Zbrojiti u perlu sve argumente naredbenog retka:  
**while($a=shift @ARGV) {$sum+=$a}**

25)Što će se u ljusci ispisati izvođenjem sljedećih naredbi:a=2;b=4;echo {$a..$b}, uz napomenu da se širenje varijabli odvija nakon širenja vitica  
**{2..4}**

j=$((7%4)); # j=3  
for i in {1..5}; #i=1;i=2;  
do [ $i -lt $j ] && sleep $i; # ispituje se da li je i<j, ako je izvršava se naredba sleep; znači niz naredbi iza && se izvršava samo ako je naredba ispred operatora znakova && vratila izlazni status "0"  
done  
  
dakle, to bi izgledalo ovako:  
1)  
j=3  
i=1  
if (i<j) sleep 1  
  
2)  
i=2  
if(i<j) sleep 2  
ukupno će se izvršavati 3 sekunde.

Iz popisa studenata studenti.txt treba izvući podatke o studentima prijelaznicima. Prepoznat ćemo ih po tome što njihov matični broj ne počinje s "0036". Popis studenata je oblika:  
Red\_br Prezime Ime (JMBAG) grupa  
U datoteci ima i redaka koji ne sadrže podatke o studentu (npr. zaglavlje tablice) i oni se ne smiju pojaviti u ispisu. Ispis podataka o studentima prijelaznicima treba imati oblik:  
JMBAG Prezime Ime  
Koja će od sljedećih skripti biti prikladna:  
  
a)  
find studenti.txt -regex '\([0-9]{10}\) && \(0036[0-9]{6}\)' | cut -f4,2,3 -d" "  
b)  
cat studenti.txt | grep -E '\([0-9]{10}\) && \(0036[0-9]{6}\)' |  
sed -r 's/[^[:alpha:]]+([[:alpha:]]\*.\*)\{([0-9]{10})\)/\2 \1/'  
c)  
grep -E '[0-9]{10}' studenti.txt | grep -E '^0036[0-9]{6}' |  
cut -f2 -d"(" | cut -f1 -d")" | cut -f2,3 -d" "  
d)  
groff -v -regex '([:digit:]\*{10})\(0036[:digit:]\*{6})' && sed -r 's/[\*]([0-9+])/$2 $1/'  
e)  
grep -E '[0-9]{10}' studenti.txt | grep -E -v '0036[0-9]{6}' |  
sed -r 's/[^[:alpha:]]+([[:alpha:]]\*.\*)\(([0-9]{10})\).\*/\2 \1/'  
f) nista od navedenog

sed -r 's/[^[:alpha:]]+([[:alpha:]]+.\*)\(([0-9]{10})\).\*/\2 \1/'  
  
[^[:alpha:]]+ == znak koji nije slovo jednom ili vise puta  
[[:alpha:]]+ == znak koji je slovo jednom ili vise puta  
.\* == bilo kakav niz znakova  
  
\(([0-9]{10})\).\* == 10 znamenki unutar zagrada pa bilo kakav niz  
  
i to ti se mijenja s ([0-9]{10}) ([[:alpha:]]+.\*)...  
  
jedino treba s .\* bit oprezan jer on "pojede" max znakova kolko moze...  
  
ako imas niz znakova:  
  
aaa111bbb222  
  
i regex:  
  
(.\*)([0-9]{3})  
-> \1 == .\* == aaa111bbb  
-> \2 == [0-9]{3} == 222  
  
pojest ce sve do 222, a ne samo do 111...  
  
dakle do najdalje do kud moze doci..jerbo je mnogo pohlepan



pa tocno to kaj pise http://www.fer2.net/images/smilies/biggrin.gif  
nije moguce napisati " \*" i ocekivati da bude ti matchal <space> \* puta.  
zato moras koristiti [ ]  
a i ovo se moglo napisati s [ ]+  
  
tak da bi ovaj program trebal napraviti sljedece:  
izbrisi sve kaj nije alpha ili razmak  
znaci ostane  
"TclTk 858 Ruby 191 Python 265 Python 312 Perl 5101"  
a onda ovaj drugi sed od toga napravi sljedece:  
za svaki niz razmaka (bilo kolko razmaka) umetni umjesto njih novi red  
pa se ova for petlja izvrsi onoliko puta kolko ima rijeci (isto bi bilo da nema toga zadnjeg seda)  
  
~~odgovor je a) 10~~  
edit: odgovor je 5 jer brojevi nisu matchani s alpha. videl sam alnum http://www.fer2.net/images/smilies/biggrin.gif

1. sed -r 's/[^[:alpha:] ]//g' - izbriši sve što nije slovo ili razmak  
dakle nakon toga imamo "TclTk Ruby Python Python Perl"  
2. sed -r 's/[ ][ ]\*/\n/g' -sve oblika "minimalno jedan razmak" pretvori u novi red  
nakon toga imamo:  
TclTk  
Ruby  
Python  
Python  
Perl  
3. za svaki i od toga uvećaj k za 1 (na početku je k=0). kako imamo 5 riječi, k će se uvećat za 1 5 puta, i na kraju je k=5

**15. zadatak** -- irendulic

**Spoiler:**

grep -E '(^.{4})(.{2}).\*[ ]\2'  
  
trazi pojavljuje li se redak sa sljedecim svojstvima:  
  
(^.{4}) - na pocetku retka su bilokoja 4 znaka  
(.{2}) - zatim su bilokoja dva znaka  
.\* - zatim je nula ili vise bilokakvih znakova  
[ ] - pa razmak  
\2 - pa se ponavljaju ona dva znaka od tri reda iznad  
  
Znaci, odgovor je d)  
If there's a bustle in your hedgerow,  
  
jer imamo na pocetku 4 znaka (If t) zatim dva znaka su (he), pa imamo neke znakove (re's a bustle in your) pa razmak i onda se onaj (he) od ranije ponavlja.  
  
al zadatak je postavljen grep -E '^.{4}(.{2}).\*[ ]\1'...  
bar ono na slikama kolko vidim...  
na isto ti ga dode...   
  
S 2 nema smisla jer bi se 2 odnosilo na 2. zagradu u regularnom izrazu, a u njemu postoji samo jedna.  
Znači jedinica se odnosi na zagradu (oblu ofc) izraz unutar koje se ponavlja.

**17. zadatak** -- by marta.m

**Spoiler:**

ls [^t]\*[12]---> izlistaj sve što ima oblik  
bilo\_što\_osim\_slova\_t pa bilo kakav znak 0 ili više puta pa ili broj 1 ili broj 2  
ili u prijevodu izlistaj sve što ne počinje slovom t a završava ili brojkom 1 ili brojkom 2  
  
ovo ovdje nije regularni izraz već širenje imena datoteka...  
hope it helps :)

|  |
| --- |
| 15. Pretpostavimo da naredba ls daje sljedeći ispis: cygutil12 cygutil14 libgcc12 libgcc14 libint21 libint24.lst termcap.dsc cygutil12.lst cygutil14.dsc libgcc12.lst libgcc14.dsc libint22.lst libint42 termcap.lst Što će se ispisati izvođenjem naredbe ls [^t]\*[12]  (a) cygutil12 libgcc12 (b) cygutil12.lst libgcc12.lst libint22.lst (c) cygutil12 libgcc12 libint21 libint42 (d) neće se ništa ispisati (e) cygutil12 cygutil12.lst libgcc12 (f) ništa od navedenoga |

zasto je odgovor c? moze netko objasniti

Jer ti ovaj izraz uz ls znači da tražiš sve fajlove koji započinju s nečim različitim od t a završavaju na ili 1 ili 2.  
  
Između njih može bit bilo šta (\*)

**18. zadatak** -- ministar

**Spoiler:**

niz znakova uokviren jednostrukim navodnicima (single-quotes) ne  
tumači se u ljusci, već se samo prosljeđuje kao niz:  
a ti imaš a='2'.5.'2', dakle ovi jednostruki se brišu i ispiše se 2.5.2, probaj pokrenut u cyqwinu pa ćeš vidjet  
pogledaj primjer u slajdovima imaš:  
$ echo ’Tekuce kazalo je $PWD’  
ISPIS:Tekuce kazalo je $PWD  
budući da su jednostruki navodnici nije izvršena ekspanzija argumenta predznačenog sa $, a da su bili dvostruki navodnici dakle:  
$ echo "Tekuce kazalo je $PWD"  
ISPIS: Tekuce kazalo je /home/korisnik  
sad se ispiše vrijednost varijable PWD...dakle kad imaš dvostruke navodnike, bilo šta da je unutar toga predznačeno sa znakom $ se širi...

**20. zadatak** -- marta.m

**Spoiler:**

for i in [1..5];  
do j=0;  
s="";  
while [ $j -lt $i ];  
do  
s=”$s j”;  
j=$((j+1));  
done  
echo $s;  
done  
  
Ja sam ga protumačila ovako, ispravite me molim vas ako griješim, jer moj ispis ima 5 znakova j, a u riješenjima su 4.  
  
i=1; j=0; s="";  
sve dok je j < i  
radi: s=j; j=1; #provjeri da li je j < i; nije; izlazi iz petlje while  
echo $s # ispisuje:j  
i=2; j=0; s="";  
sve dok je j < i  
radi: s=j; j=1; #provjeri da li je j < i; je, ostani u while petlji  
s=j j; j=2; # dakle sam smo u s imali od gornjeg reda j, i još smo dodali razmak i j; provjeri da li je j < i; nije; izlazi iz while petlje  
echo $s # ispisuje:j j  
i=3;j=0; s="";  
...... #while petlja se izvršava tri puta  
ispisuje:j j j  
i=4;j=0; s="";  
...... #while petlja se izvršava četri puta  
ispisuje:j j j j  
i=5;j=0; s="";  
...... #while petlja se izvršava pet puta  
ispisuje:j j j j j   
  
pitaju te koji je jedan od ispisa...  
dakle tocni odgovori bi bili:  
  
j  
j j  
j j j  
j j j j  
j j j j j  
  
a jedini ponudeni je ovaj s 4 komada pa je to tocno...

|  |
| --- |
| *8. Što će se ispisati na ekranu kada se izvrši sljedeći niz naredbi? uvjet="utorak je pravi dan za ispit" case $uvjet in utorak ) echo "prvi";; .\*ispit ) echo "drugi";; \*pravi\* ) echo "treci";; [^uto]uto\* ) echo "cetvrti";; \*dan ) echo "peti";; esac   zar ne bi ovo trebalo ispisati "drugi", a ne "treci"?* |

U caseu se koristi globbing pa taj .\*ispit zapravo traži izraz "točka + proizvoljan niz znakova + ispit". Ulogu točke u regexu bi tu obavljao upitnik.  
  
Nisam ni ja znao, sad sam pogledao.

MI 2012/2013 TOCNI ODGOVORI

1.B  
2.A  
3.C  
4.C  
5.D  
6.D  
7.E  
8.A  
9.A  
10.A  
11.C  
12.B  
13.A  
14.B  
15.C  
16.A  
17.C  
18.E