**SKRJEZ – ZI**

BASH

1) Ispisati sve datoteke iz tekudeg kazala i svih podkazala koje imaju bar 2 samoglasnika u

imenu. Rješenje: ima ih svakakvih, nisam sve ponuđene zapisao. Ugl, ovo je točno:

**find . –regex “.\*\*aeiouAEIOU+.\*\*aeiouAEIOU+.\*“ –print**

* Objašnjenje: sve piše u šalabahteru. Find traži u zadanom direktoriju i njegovim poddirektorijima, a to je ovdje točka što predstavlja tekudi direktorij. Opcija regex služi baš onome što zadatak traži... da se nađu samo datoteke kojima ime odgovara danom regularnom izrazu.

2) Koji je sadržaj varijable $str nakon sljedećeg odsječka:

my $str = '112133';

$str =~ s/(.)\1/$1/g;

* U varijabli $str nakon izvršavanja nalaziti će se string '**1213**'. Izvršit de se zamjena bilo koja 2 uzastopna ista znaka sa samo jednim tim istim znakom... i to nad varijablom $str (bindanje).

3) Ima hrpa nekih imena datoteka, pitanje je koja od njih de ispisati sljededa linija koda u

bash-u: **ls [^CA]\*[NB]?.t\***

* Ovo ispisuje sve datoteke u tekudem direktoriju čiji naziv: NE počinje sa 'C' ili 'A' iiiii koje u sebi sadrže podniz: **(N ili B)(jedan bilokoji znak).t** . Ono čega se morate sjetiti je da ovo nije regularni izraz nego sintaksa bash-a, a u njoj \* predstavlja isto što i .\* u regexu, a ? predstavlja isto što i . u regexu.

4) Kako u bash-u evaluirati izraz (e-4)/d i pridružiti rezultat varijabli a.

* Odgovor: **$(( ($e-4)/$d ))** Izrazi se evaluiraju u dvostrukim zagradama prethođenim dolar-znakom. Usput samo da kažem... unutar tih $(( )) izraza se mogu izostavljati $ pred imenom varijabli. Dakle *$(( (e-4)/d))* je jednako dobar izraz.

5) umjesto sa stdin, neka program čita datoteku:

* program <datoteka

To je operator, sto da ti kazem.

6) koji popis datoteka se može dobiti s: echo [^ain]\*im\*

* Nepotpuno pitanje, ali valjda sve datoteke koje ne počinju sa jednim od slova (a, i, n) a imaju niz "im" u sebi.

7) znamo da imamo neki python file koji importa t...(?), i neznamo je li prvo t veliko ili malo, koja naredba će naći takve datoteke:

* grep -rE 'import [tT]...' .

-r -> Rekurzivno grepaj kroz direktorije

-E -> Extended regularni izrazi

izraz: 'import [Tt].+'

Počinje sa import, onda ide T ili t, zatim bilo što, bar jednom.

8) Naci datoteke vece od 2 mb a manje od 16

* find /bin -size +2 -a -size -16

9) ls vrati imena datoteka, prosirivanje niza u bashu \*[^NBI].\*, sto ostane

* 3 datoteke, arial.tff i jos dvije koje nemaju N, B, ili I ispred .

10) $a[3]=3; $a[7]=7; $a[8]=8; $a=$a[0]+1; print $a;

sto je u varijabli $a?

a) undef

* **b) 1**

c) 0

...

f) nista od navedenog

PERL

1) Imamo datoteku koja sadrži retke oblika: “login“;“Ime“;“Prezime“;JMBAG;“email“.

Nadopuniti skriptu da joj izlaz bude: JMBAG Prezime Ime

While(<>){

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ }

a) s/(.\*);/\4\™\3 \2\n/g){print $\_;} – ovo je bugovito do bola...

b) if (@a= /;\“(.\*)\“;/),print “@a\*3+\t ...“- – nema opcije „g“ kod matchanja

* c) **if(@a= /\“\*(.\*?)\“\*;/g),print “$a\*3+\t $a[2] $a[1]\n“;-** - ovo je točno, iako je užasno čudan regex... puno toga nepotrebnog i zbunjujudeg

d) @a = $\_.split(';'); ... – nije ovo python xD Al skoro me zeznulo xD

e) If (/;(.\*);/g),print “\3\t\2 \1\n“;- – tako se ne koristi backreference, trebaju biti $1 itd.

2) $string = “To dream od the person you would like to be is to waste the person you are.\n“;

@a = $string =~ /\b(\w{5,})\s/g;

* Ovo će vratiti u listu @a sve podnizove u varijabli $string koji su dugački barem 5 znakova i započinju znakom \b (označava početak riječi) i završavaju nekim praznim znakom (znači razmak ili \n ili štogod...). To znači da de lista @a imati 5 članova: **dream, person, would, waste, person.**

3) Što će se ispisati nakon sljedećeg prog. odsječka?

@a = (1, 2, 3, 4);

@b = (5, @a, 6, 7);

%c= %b; print %c . „ : „ . $b[1];

* 7 : 1

4) argumenti potprograma... : $\_[0]+$\_[1]

* Argumenti se nalaze u polju @\_, ovo je sintaksa za pristupanje.

5) Kako bi iz stringa a izvadili od 4 do 10 znaka?

* @l = a=~ /.{4}(.{6})/; $l[0] tako nešta... drugi odgv su bili sa slajsanjem a[4]... ali string se u prelu nemože slajsati
* Kada u perlu spremis rezultat primjene regexa u listu dobijes listu grupa (kad koristis match, barem). Prva 4 znaka ne spremas u grupu, a onda spremas ostalih 6, koji idu u grupu 0 (tj. grupu 1 kad radis swap, ali 0. element)

6) @l = (3,5,8,7,9,3,6,7,8,9); %e = reverse @l; print $e{7} #8

* Ako okrenes dobijes listu. (9, 8, 7, 6, 3, 9, 7, 8, 5, 3). Ali, buduci da to spremamo u dict, svaki prvi postaje kljuc, a svaki drugi vrijednost, dakle: (9 => 8, 7 => 6, 3 => 9, 7 => 8, 5 => 3) Imamo dvaput 7, ali hashevi rade na prebrisivanje, dakle uzimamo drugi, ergo vrijednost na kljucu 7 je 8.

7) $a=3; foreach (1..7,7..5){print($a+=$\_);} print ($a) #4691318243131

* U perlu $a += element vraca vrijednost koja se ispisuje.

a = 3 + 1 = 4

a = 4 + 2 = 6

a = 6 + 3 = 9

a = 9 + 4 = 13

a = 13 + 5 = 18

a = 18 + 6 = 24

a = 24 + 7 = 31

Opet ispisuje a, dakle 31

E, ali zašto ne 7..5? Pa spajamo rangeove. E, pa ako je desna strana veća od lijeve, stvara se prazna lista.

8) onaj koji je netko ovdje pitao: @a=(0,1,2); my ($b) = @a; print $b # 0

* Zbog zagrada, $b je u kontekstu liste. Stoga u njega će se spremiti prvi element. Da imamo npr.

my ($b, $c) = @a

Onda bi $b = 0, $c = 1

9) Kako kružno zamijeniti vrijednosti varijabli –

* ($a,$b,$c) = ($b,$c,$a)

Staro pitanje. Valjda jasno?

10) @lista = (1,2,3,4); ($f,$d,$e) = @lista; @lista = ($lista[2], $f ,@lista[$d], $#lista) # (3,1,3,3) -promjenio sam kako je bilo u mojoj grupi C

* ($f, $d, $e) = (1,2,3)

@lista = ($lista[2], $f, @lista[$d], $#lista) = (lista\_na\_index\_2, $f, $lista[2], indeks\_zadnjeg\_clana\_liste) = (3, 1, 3, 3)

11) Neka recenica, regularni izraz: "\b(\w{4,})\s", koliko vrijednosti vrati izraz:

* 6, pocetak rijeci, 4 ili vise slova, zavrsava s whitespace znakom, znaci ne zarez

12) Iz niza znakova dobiti od 5. do 10. znaka u novi niz.

* ~/\w{4}(\w{6})/, perl nemoze indeksirati znakovne nizove

13) Hash $h{'r'}=4, $h{4}=5, print $h{3}.$h{4} :

* 5

14) Znakovni niz ima obli "Grupa";"Ime";"Prezime";"JMBAG";"Bodovi", dobiti ispis oblika JMBAG Prezime Ime

* @a = ($s =~ m/"(.\*?)";/g ); print "$a[3]\t$a[2] $a[1]";

15) (od kolege mmatija) @l = (3,5,8,7,9,3,6,7,8,9); %e = reverse @l; print $e{7}

* 8

16) (od kolege mmatija) $a=3; foreach (1..7,7..5){print($a+=$\_);} print ($a)

* 4691318243131

17) (od kolege mmatija) @a=(0,1,2); my ($b) = @a; print $b

* 0

18) @a = qw / ana branka dragica ana tea jana /;

foreach (@a) { @b=/./g; $c{$\_}=@b; }

foreach (values (%c)) { $s+=$\_; }: print $s;

sto ce se ispisati na zaslonu :

a) 26

* **b) 23**

c) ana branka dragica tea jana

...

f) nista od navedenog