

C Uվшզшն Shell 00

<ակիրճ. այս փաստաթուղթը 42 դպրոցի C ավազանի Shell 00 մոդուլի նյութն t:

Ցանկ

1	Յուցուսներ	2
II	Նախաբան	4
III	Առաջադրանք 00։ Z	6
IV	Առաջադրանը 01։ testShell00	7
\mathbf{v}	Առաջադրանք 02։ Այո՜, էլիիի՜/ ավելիիի՜ն	8
VI	Առաջադրանը 03։ SSH ինձ	10
VII	Առաջադրանը 04։ midLS	11
VIII	Առաջադրանը 05։ GiT commit	12
IX	Առաջադրանը 06։ gitignore	14
X	Առաջադրանք 07։ diff	15
XI	Առաջադրանք 08։ Չնջել	16
XII	Առաջադրանը 09։ Պատրանը, ոչ թե հնարը, Միքայե՜լ	17
XIII	Հանձնում և ընկերն ընկերոջը ստուգում	18

Գլուխ I

Յուցումներ

- Այս էջը ձեր միակ ուղեցույցն է։ Պատվող խոսակցություններին ուշադրություն մի՛ դարձրեք։
- Ձգուշացում. մինչ առաջադրանքները հանձնելը նորից ստուգե՜ք նյութը։ Յանկացած պահի այս փաստաթուղթը կարող է փոփոխվել։
- Ուշադրություն դարձրե՛ք ձեր ֆայլերի և պահոցների թույլտվություններին։
- Բոլոր առաջադրանքները կատարելիս անհրաժեշտ է հետևել հանձման ընթացակարգին։
- Ձեր առաջադրանքները կստուգվեն ձեր դասընկերների կողմից։
- Քացի դրանից, ձեր առաջադրանքները կստուգվեն և կգնահատվեն Moulinette կոչվող ծրագրով։
- Moulinette-ը գնահատելիս շատ բծախնդիր է ու խիստ։ Այն ամբողջովին ավտոմատացված է, և գնահատման հարցում նրա հետ անհնար է բանակցել։ Այսպիսով, տհաճ անակնկալներից խուսափելու համար առաջադրանքները պետք է կատարվեն հնարավորինս անթերի։
- Moulinette-ն այնքան էլ լայնախոհ չէ։ Այն չի էլ փորձի հասկանալ ձեր կոդը, եթե վերջինս չի համապատասխանում Norm-ին։
- Մուլինետի աշխատանքը հիմնված է norminette կոչվող ծրագրի վրա, որը ստուգում է, թե արդյոք ձեր ֆայլերը համապատասխանում են Norm-ին։ Կարճ ասած, norminette-ի ստուգման թեստը չանցած աշխատանքը չի ընդունվի։
- Առաջադրանքները դասավորված են ըստ բարդության աստիճանի` ամենապարզից ամենաբարդը։ Հաջողությամբ կատարված բարդ առաջադրանքները հաշվի չեն առնվի եթե պարզ առաջադրանքներից որևէ մեկը լիարժեք չի աշխատում։
- Արգելված ֆունկցիաների կիրառումը համարվում է խարդախություն։ Խարդախությունը պատժվում է -42-ով, և այս գնահատականը քննարկման ենթակա չէ։

- main () ֆունկցիա պետք է հանձնել միայն այն դեպքում, եթե պահանջվի գրել ծրագիր։
- Moulinette-ը կազմարկում է այս դրոշակների օգնությամբ` Wall Wextra -Werror, և գործածում է gcc:
- Եթե ձեր ծրագիրը չկազմարկվի, կստանաք 0։
- Նյութում նշված ֆայլից բացի ձեր պահոցում <u>հավելյալ</u> ֆայլեր չպետք է լինեն։
- Հարցեր կա՞ն։ Դիմե՛ք աջ կողմում նստած դասընկերոջը։ Կամ Էլ դիմե՛ք ձախ կողմինին։
- Ձեր ուղեցույցներն են Google-ը/ man / hամացանցը / ...։
- Կարող եք օգտվել նաև ներքնացանցի ֆորումի "C Piscine" հատվածից կամ slack Piscine-ից։
- Մանրակրկիտ ուսումնասիրե՜ք օրինակները։ Շատ հնարավոր է, որ դրանք պահանջեն նյութում հստակորեն չնշված մանրամասներ։
- Հանուն Օդինի, հանուն Արամազդի։ Ուղեղներդ ի գո՛րծ։

Գլուխ II Նախաբան

«Քաղաքի որսորդները» ֆիլմում հնչող երգի` «Լուսնկայի ստվեր»-ի բառերը։

Վերջին անգամ նրան տեսավ` Լուսնկայի ստվերով տարված, Անցավ նա անհանգիստ ու զգուշատես՝ Լուսնկայի ստվերով տարված։ Շաբաթ գիշերվա հանելուկում կորած՝ Հեռու՜- հեռու՜ հակառակ կողմում, Ընկավ նա կովի մեջ հուսահատ. Իսկ ինքը չիմացավ՝ ինչպես առաջ շարժվել։ Ծառերը, որ շշնջում են գիշերում՝ Լուսնկայի ստվերով տարված, Երգր երգի՛ր վիշտ ու ցավի՝ Լուսնկայի ստվերով տարված։ Նա տեսավ միայն ստվերը զենքի Հեռու´-հեռու´` հակառակ կողմում։ Կրակեցին վեց անգամ, Իսկ ինքը չիմացավ՝ ինչպես առաջ շարժվել։

[Կրկներգ]

Ես կմնամ, ես կաղոթեմ, Կհանդիպենք հեռավոր դրախտում, Ես կմնամ, ես կաղոթեմ, Կհանդիպենք մի օր դրախտում։

Առավոտյան ժամը չորսին` Լուսնկայի ստվերով տարված, Ես` ականատես քո տեսիլքին` Լուսնկայի ստվերով տարված, Աստղ շողացով գիշերվա երկնքում, Հեռու´-հեռու´ հակառակ կողմում։ Կգա՞ս խոսենք մենք այս գիշեր, С Ավազան

Shell 00

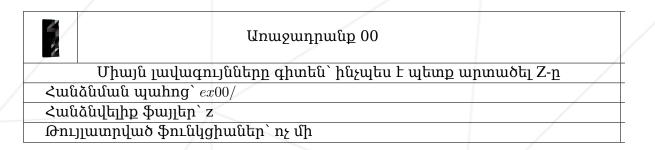
Քայց ինքը չգիտեր՝ ինչպես առաջ շարժվել։

[Կրկներգ]

Հեռու՜-հեռու՜ հակառակ կողմում՝ Ընկած հարյուր հինգի միջև, Գիշերը ծանր էր իսկ օդը՝ թեթև, Բայց ինքը չիմացավ` ինչպես առաջ շարժվել։ Լուսնկայի ստվերով տարված՝ Լուսնկայի ստվերով տարված՝ Հեռու՜-հեռու՜ հակառակ կողմում։

Դժբախտաբար, այս թեման *«Քաղաքի որսորդներ»*-ի հետ ոչ մի կապ չունի։

Գլուխ III Առաջադրանք 00։ Z

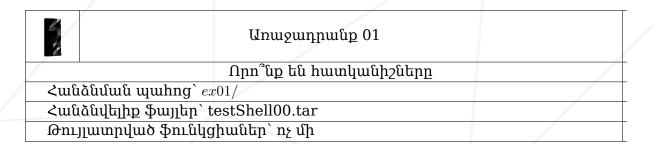


• Ստեղծել z անունով ֆայլ, որը վերադարձնում է "Z"` հաջորդված նոր տողով, երբ ֆայլի վրա գործածվում է cat հրամանը։

?>cat z Z

Գլուխ IV

Առաջադրանք 01: testShell00



- Ստեղծել testShell00 անունով ֆայլ ձեր հանձնման պահոցում։
- Գտնել ե<u>լքը</u> այսպես արտածելու մի տարբերակ ("total 1" տողից բացի)։

```
%> ls -l
total 1
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

• Նախորդ քայլերը ավարտելուն պես պետք է գործարկել tar -cf testShell00.tar testShell00 հրամանը՝ հանձնվելիք ֆայլը ստեղծելու համար։



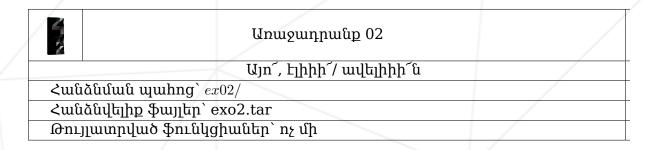
Մի´ անհանգստացեք այն մասին, թե ինչ եք ստացել "XX"-ի փոխարեն



Որպես պատասխան` ժամի փոխարեն կընդունվի տարեթիվը։

Գլուխ V

Առաջադրանք 02։ Այո՜, Էլիիի՜/ ավելիիի՜ն



• Ստեղծել հետևյալ ֆայլերն ու պահոցները։ Այնպես արեք, որ ձեր պահոցում 1s –1 հրամանը կատարելիս ելքը լինի այսպիսին`

```
%> ls -l
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x---r-- 2 XX XX XX Jun 1 22:45 test2
-r----r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r---x 1 XX XX 2 Jun 1 23:43 test4
-r------ 2 XX XX 1 Jun 1 23:44 test5
lrwxrwxrwx 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

• Ավարտելուն պես աշխատեցրեք tar -cf exo2.tar * հրամանը` հանձնվելիք ֆայլը ստեղծելու համար։



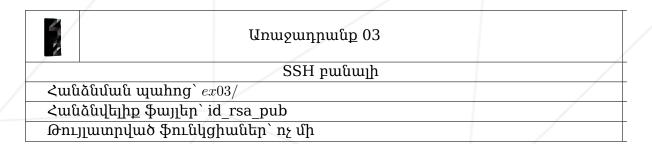
Մի´ անհանգստացեք այն մասին, թե ինչ եք ստացել "XX"-ի փոխարեն



Որպես պատասխան` կընդունվի մեկ տարին ժամի փոխարեն։

Գլուխ VI

Առաջադրանք 03։ SSH ինձ



- Ստեղծե՛ք ձեր SSH բանալին, այնուհետև հետևեք քայլերին։
 - Ավելացրե´ք ձեր public բանալին պահոցում, id_rsa_pub պանակում։
 - Թարմացրե՛ք ձեր ssh բանալին ինտրանետում։ Սա ձեզ թույլ կտա ձեր պահոցը պահպանել git սերվերում։



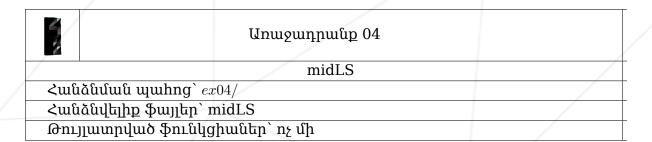
Ֆայլի անունը պատահականորեն չի ընտրվել։



Վստահ եղե՛ք, որ հասկացել եք public և private բանալիների տարբերությունը։

Գլուխ VII

Առաջադրանք 04։ midLS



• midLS ֆայլում տեղադրե՛ք հրաման-տողը, որը կթվարկի ձեր ներկայիս պահոցի բոլոր ֆայլերն ու պահոցները ` (բացառությամբ թաքնված կամ .-ով սկսվող ֆայլերի, այո, ներառյալ երկու կետով սկսվողները) իրարից ստորակետերով և բացատներով անջատված, ըստ խմբագրման ամսաթվի դասավորված։ Համոզվե՛ք, որ պահոցների անուններից հետո շեղ գծի (slash) նշան է դրված։



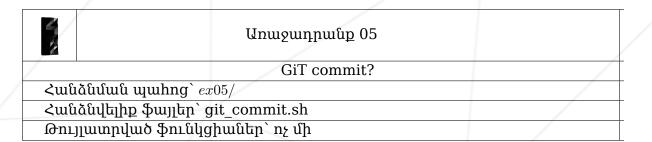
Այն, ինչ, որ չի պահանջվել, հարկավոր չի անել



Կարդացե՛ք գրողի տարած ձեռնարկը (man)։

Գլուխ VIII

Առաջադրանք 05։ GiT commit



• Ստեղծել shell script, որը արտածում է ձեր git պահոցի վերջին 5 commit-ների id-ները։

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80eea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```

Ձեր script-ը ստուգելու համար գործածելու ենք մեր սեփական միջավայրը։



Կարդացե՛ք գրողի տարած ձեռնարկը (man)

Հասաք նշաձողին։ Շարունակե՜ք։

Դուք հասաք այս նախագիծը հաջողելու համար պարտադիր առաջադրանքների ավարտին։

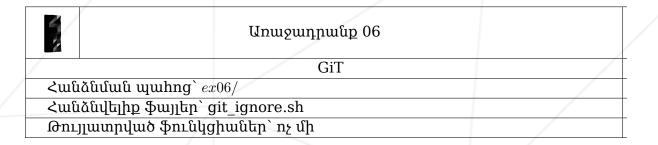
Կարող եք ինքներդ որոշել` ցանկանում եք արդյոք շարունակել և կատարել հետևյալ կամընտրական առաջադրանքները, թե անցնել հաջորդ նախագծին։ Վաղ թե ուշ կտեսնեք, որ երկու ուղիներն էլ օգտակար են լինելու։

Ընտրություն կատարելու համար հաշվի առեք հետևյալ կետերը.

- Ձեր ամենաառաջին քննությունը լինելու է C-ով ծրագրման մասին։ Մինչ այդ, հավանաբար, արդեն կառնչվեք առաջին C նախագծի հետ։ Նույնը վերաբերում է Rush նախագծերին, որոնք կատարելու եք շաբաթվա վերջում (դրանց մասին շուտով ավելին կիմանաք)։
- Ձեր գերազանցությունը այս ավազանում գնահատվելու է բազմաթիվ գործոններով։ Յուրաքանչյուր ծրագիր ավարտին հասցնելը դրանցից մեկն է, բայց ընդհանուր առաջընթացն ավազանի բոլոր նախագծերում մեկ ուրիշ գործոն է։ Ընտրեք խելամտորեն՝ ձեր արդյունքները օպտիմալացնելու համար։
- Միշտ հնարավոր կլինի նույն նախագիծը վերահանձնել մի քանի օր/շաբաթից` մինչև ավազանի ավարտը։
- Ձեր ընկերների հետ համաժամեցված մնալը հնարավորթյուն է տալու նրանց հետ ավելի լավ համագործակցել։

Գլուխ IX

Առաջադրանք 06: gitignore



• Այս առաջադրանքում պետք է գրել մի կարճ shell սկրիպտ, որը ցուցակագրում է ձեր GiT պահոցի կողմից անտեսված բոլոր ֆայլերը։ Օրինակ՝

```
%> bash git_ignore.sh | cat -e
.DS_Store$
mywork.c-$
%>
```

Ձեր script-ը ստուգելու համար գործածելու ենք մեր սեփական միջավայրը։



Կարդացե՛ք գրողի տարած ձեռնարկը (man)

Գլուխ X

Առաջադրանք 07։ diff



Առաջադրանք 07

<անձնման պահոց` ex07/</br>
<անձնվելիք ֆայլեր` b

Թույլատրված ֆունկցիաներ` ոչ մի

• Ստեղծել ь ֆայլ այնպես որ`

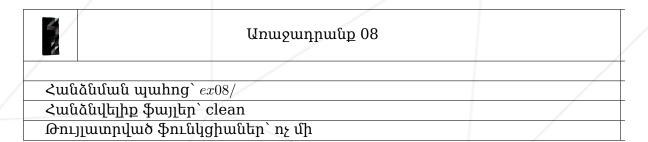
%>diff a b > sw.diff



man patch

Գլուխ XI

Առաջադրանք 08։ Ձնջել



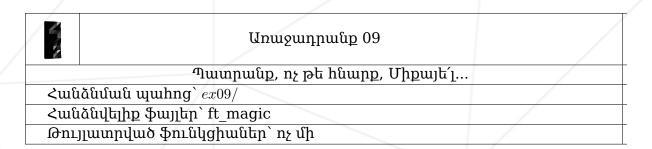
- clean անունով ֆայլում դնել այն հրաման-տողը, որը ներկայիս պահոցում, ինչպես նաև դրա ենթապահոցներում կփնտրի ~-ով վերջացող կամ #-ով սկսվող ու վերջացող անուններով ֆայլերը։
- Հրաման-տողը ցույց կտա ու կջնջի բոլոր գտնված ֆայլերը։
- Միայն մի հրաման է թույլատրվում` ոչ մի ';' կամ '&&' կամ այլ հիմար խարեբայություններ։



man find

Գլուխ XII

Առաջադրանք 09։ Պատրանք, ոչ թե հնարք, Միքայե՜լ...



• Ստեղծել ft_magic անունով մի կախարդական ֆայլ, որի ձևաչափը կառուցված կլինի այնպես, որ կարողանա գտնել 42 file տեսակի ֆայլերը` կառուցված "42" տողով, 42-րդ բայթում։



man file



Ձախողելը սովորելու մի մասն է։

Գլուխ XIII

<անձնում և ընկերն ընկերոջը ստուգում

Հանձնեք ձեր առաջադրանքը Git պահոցում, ինչպես սովորաբար անում եք։ Ստուգման ժամանակ գնահատվելու է միայն ձեր պահոցի պարունակությունը։ Մի՛ վարանեք նորից ստուգել ձեր ֆայլերի անունները՝ համոզվելու համար, որ դրանք ճիշտ են։



Հարկավոր է հանձնել միայն այն ֆայլերը, որոնք պահանջվում են այս նախագիծը նկարագրող ֆայլում։