



Շ Ավագան

Շ 04

Հակիրճ. այս փաստաթուղթը 42 դպրոցի Շ ավագանի Շ 04 մոդուլի
նյութն է:

Ցանկ

I	Ցուցումներ	2
II	Նախաբան	4
III	Առաջադրանք 00 : ft_strlen	6
IV	Առաջադրանք 01 : ft_putstr	7
V	Առաջադրանք 02 : ft_putnbr	8
VI	Առաջադրանք 03 : ft_atoi	9
VII	Առաջադրանք 04 : ft_putnbr_base	10
VIII	Առաջադրանք 05 : ft_atoi_base	12
IX	Հանձնում և ընկերն ընկերոջը ստուգում	13

Գլուխ I

Ցուցումներ

- Այս էջը ձեր միակ ուղեցույցն է: Պատվող խոսակցություններին ուշադրություն մի՛ դարձրեք:
- Չգուշացում. մինչ առաջադրանքները հանձնելը նորից ստուգե՛ք նյութը: Ցանկացած 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:
- Բացի 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, տիպիկ անակնկալներից խուսափելու համար առաջադրանքները պետք է կատարվեն հնարավորինս անթերի:
- Moulinette-ն այնքան էլ լայնախոհ չէ: Այն չի էլ փորձի հասկանալ ձեր կողը, 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 `cc`:
- Եթե ձեր ծրագիրը չկազմարկվի, կստանաք 0:
- Նյութում նշված ֆայլից բացի ձեր պահոցում հավելյալ ֆայլեր չպետք է լինեն:
- Հարցեր կա՞ն: Դիմե՛ք աջ կողմում նստած դասընկերոջը: Կամ էլ դիմե՛ք ձախ կողմինին:
- Ձեր ուղեցույցների անուններն են *Google/ man/ internet/ ...*
- 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:
- Դե՛, ձեզ տեսնենք: Հանուն Օդինի, հանուն Արամազդի: Ուղեղներդ ի գո՛րծ:



Norminette-ը պետք է գործարկվի `-R CheckForbiddenSourceHeader` դրոշակով: Այն նաև անհրաժեշտ է Moulinette-ի աշխատանքի համար:

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:
Ծաբաթ գիշերվա հանելուկում կորած՝
Հեռու՝-հեռու՝ հակառակ կողմում,
Ընկավ նա կովի մեջ հուսահատ,
Իսկ ինքը չիմացավ՝ ինչպես առաջ շարժվել:
Ծառերը, որ շնչում են գիշերում՝
Լուսնկայի ստվերով տարված,
Երգը երգի՛ր վիշտ ու ցավի՝
Լուսնկայի ստվերով տարված:
Նա տեսավ միայն ստվերը գեների
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,

Բայց 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 : ft_strlen


	Առաջադրանք 00
	ft_strlen
	Հանձնման պահոց՝ ex00/
	Հանձնվելիք ֆայլեր՝ ft_strlen.c
	Թույլատրված ֆունկցիաներ՝ ոչ մի

- Ստեղծել ֆունկցիա, որը հաշվում և վերադարձնում է տողի նիշերի քանակը :
- Նախատիպը պետք է լինի այսպիսին՝

```
int  ft_strlen(char *str);
```

Գլուխ IV

Առաջադրանք 01 : ft_putstr


	Առաջադրանք 01
	ft_putstr
	Հանձնման պահոց՝ ex01/
	Հանձնվելիք ֆայլեր՝ ft_putstr.c
	Թույլատրված ֆունկցիաներ՝ write

- Ստեղծել ֆունկցիա, որը տողի նիշերը արտածում է էկրանին:
- Տողի առաջին նիշի հասցեն ֆունկցիային որպես պարամետր տրված ցուցիչն է:
- Նախատիպը պետք է լինի այսպիսին՝

```
void    ft_putstr(char *str);
```


Գլուխ V

Առաջադրանք 02 : ft_putnbr

	Առաջադրանք 02
	ft_putnbr
	Հանձնման պահոց՝ ex02/
	Հանձնվելիք ֆայլեր՝ ft_putnbr.c
	Թույլատրված ֆունկցիաներ՝ write


- Ստեղծել ֆունկցիա, որն արտածում է որպես պարամետր տրված թիվը: Ֆունկցիան պետք է կարողանա արտածել int տեսակի ցանկացած փոփոխական:
- Նախատիպը պետք է լինի այսպիսին՝

```
void ft_putnbr(int nb);
```

- Օրինակ՝
 - ft_putnbr(42)-ը արտածում է "42":

Գլուխ VI

Առաջադրանք 03 : ft_atoi

	Առաջադրանք 03
	ft_atoi
	Հանձնման պահոց՝ ex03/
	Հանձնվելիք ֆայլեր՝ ft_atoi.c
	Թույլատրված ֆունկցիաներ՝ ոչ մի

- Ստեղծել ֆունկցիա, որը տողի str-ով ցույց տված առաջին մասը փոխակերպում է դրա int ներկայացման:
- Տողը կարող է սկսվել ցանկացած քանակությամբ բացատներով (ինչպես որոշված է isspace(3) հրահանգով):
- Տողի վերջում կարող են լինել ցանկացած քանակի + և - նշաններ, - նշանը կփոխի վերադարձված int-ի նշանը՝ կախված - նշանների կենտ կամ գույգ քանակից:
- Վերջապես, տողի վերջում կարող է լինել ցանկացած 10 հիմքով թիվ:
- Ֆունկցիան պետք է կարդա տողը, քանի դեռ այն բավարարում է պայմաններին և վերադարձնի մինչև այդ գտնված թիվը:
- Գերհագեցմանը կամ թերհագեցմանը հարկավոր չէ ուշադրություն դարձնել: Այդ դեպքերում արդյունքը կարող է լինել անորոշ:
- Ահա ծրագրի մի օրինակ, որը տպում է atoi-ի վերադարձված արժեքը՝


```
$>./a.out " ----++1234ab567"  
-1234
```

- Նախատիպը պետք է լինի այսպիսին՝

```
int ft_atoi(char *str);
```

Գլուխ VII

Առաջադրանք 04 : ft_putnbr_base

	Առաջադրանք 04
	ft_putnbr_base
	Հանձնման պահոց՝ ex04/
	Հանձնվելիք ֆայլեր՝ ft_putnbr_base.c
	Թույլատրված ֆունկցիաներ՝ write


- Ստեղծել ֆունկցիա, որը տերմինալում արտածում է թիվը՝ անհրաժեշտ հիմքով համակարգում ներկայացված:
- Այս թիվը տրվում է int տեսքով, իսկ հիմքային թիվը *նիշերի տողի տեսքով*:
- Հիմքային համակարգը պարունակում է բոլոր գործածելի սիմվոլները այդ թիվը ներկայացնելու համար:
 - 0123456789-ը տասական թվերի ներկայացման հիմնականում գործածվող համակարգն է:
 - 01-ը երկուական համակարգ է:
 - 0123456789ABCDEF-ը տասնվեցական համակարգ է:
 - poneyvif-ը ութական համակարգն է:
- Ֆունկցիան պետք է կարողանա բացասական թվերի հետ աշխատել:
- Եթե արգումենտը անվավեր է, ոչինչ չի արտածվի: Անվավեր արգումենտների օրինակներ՝
 - հիմքը դատարկ է կամ չափը մեկ է,
 - հիմքում նույն նիշը կրկնվում է երկու անգամ,

- հիմքը պարունակում է + կամ - :
- Նախատիպը պետք է լինի այսպիսին`

```
void ft_putnbr_base(int nbr, char *base);
```

Գլուխ VIII

Առաջադրանք 05 : ft_atoi_base

	Առաջադրանք 05
ft_atoi_base	
Հանձնման պահոց՝ ex05/	
Հանձնվելիք ֆայլեր՝ ft_atoi_base.c	
Թույլատրված ֆունկցիաներ՝ ոչ մի	

- Ստեղծել ֆունկցիա, որը տողի str-ով ցույց տված առաջին մասը փոխակերպում է int ներկայացման:
- str-ն ներկայացված է երկրորդ պարամետրով տրված հիմքով:
- Բացառությամբ հիմքի կանոնի՝ ֆունկցիան պետք է աշխատի ճիշտ ft_atoi ֆունկցիայի պես:
- Անվավեր արգումենտի դեպքում ֆունկցիան կվերադարձնի 0:
Անվավեր արգումենտների օրինակներ՝
 - հիմքը դատարկ է կամ չափը մեկ է,
 - հիմքում նույն նիշը կրկնվում է երկու անգամ,
 - հիմքը պարունակում է + կամ - կամ բացատներ:
- Նախատիպը պետք է լինի այսպիսին՝

```
int ft_atoi_base(char *str, char *base);
```

Գլուխ IX

Հանձնում և ընկերն ընկերոջը ստուգում

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



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