

C Ավազան C 05

<ակիրճ. այս փաստաթուղթը 42 դպրոցի C ավազանի C 05 մոդուլի նյութն t:

Ցանկ

1	ənigniuuli	2
II	Նախաբան	4
ш	Առաջադրանք 00 ։ ft_iterative_factorial	6
IV	Առաջադրանք 01 ։ ft_recursive_factorial	7
\mathbf{v}	Առաջադրանք 02 ։ ft_iterative_power	8
VI	Առաջադրանք 03 ։ ft_recursive_power	9
VII	Առաջադրանը 04 ։ ft_fibonacci	10
VIII	Առաջադրանք 05 ։ ft_sqrt	11
IX	Առաջադրանք 06 ։ ft_is_prime	12
X	Առաջադրանք 07 ։ ft_find_next_prime	13
XI	Առաջադրանք 08 ։ Տասը թագուհիներ <u>ը</u>	14
XII	Հանձնում և ընկերն ընկերոջը ստուգում	16

Գլուխ I

Յուցումներ

- Այս էջը ձեր միակ ուղեցույցն է։ Պտտվող 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։
- 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 cc:
- 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 ուղեցույցների անուններն են Google/ man/ internet/ ...
- Կարող եք օգտվել նաև ներքնացանցի ֆորումի «C Piscine» hատվածից 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

Հավանաբար, հետևյալ կատակը կօգնի ավելի հեշտ կատարել առաջադրանքները։

Ոստիկանը հարցաքննում է կասկածյալին։

ՈՍՏԻԿԱՆ.- Պարո՜ն Պետրոսյան, որտեղի՞ց Ձեզ այդքան գումար։

ՊԵՏՐՈՍՅԱՆ.- Մեր տան դարակից։

ՈՍՏԻԿԱՆ.- Իսկ ո՞վ է գումարը դրել դարակի մեջ։

ՊԵՏՐՈՍՅԱՆ.- Կինս։

ՈՍՏԻԿԱՆ.- Իսկ Ձեր կնո՞ջն ով է տվել այդ գումարը։

ՊԵՏՐՈՍՅԱՆ.- Ես եմ տվել։

ՈՍՏԻԿԱՆ.- Իսկ որտեղի՞ց Ձեզ այդքան գումար։

ՊԵՏՐՈՍՅԱՆ.- Մեր տան դարակից։

ՈՍՏԻԿԱՆ.- Իսկ ո՞վ է գումարը դրել դարակի մեջ։

ՊԵՏՐՈՍՅԱՆ.- Կինս։

ՈՍՏԻԿԱՆ.- Իսկ Ձեր կնո՞ջն ով է տվել այդ գումարը։

ՊԵՏՐՈՍՅԱՆ.- Ես եմ տվել։

ՈՍՏԻԿԱՆ.- Իսկ որտեղի՞ց Ձեզ այդքան գումար։

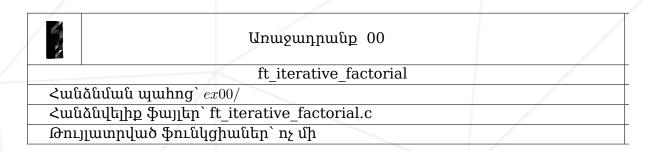
•••••

Հիմա ես գիտեմ, որ դու արդեն գիտես, թե ինչ բան է ռեկուրսիան։ Ես

С Ավազան C 05 նաև գիտեմ, որ դու գիտես, որ ես գիտեմ, որ դու գիտես, որ ես գիտեմ, որ դու գիտես, ..., թե ինչ է ոեկուրսիան։ 5

Գլուխ III

Unugunpuup 00: ft_iterative_factorial

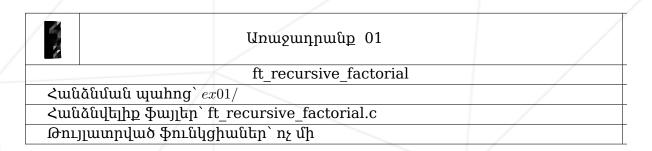


- Ստեղծել իտերացիոն ֆունկցիա, որը վերադարձնում է թիվ։ Այդ թիվը որպես մուտքային պարամետր փոխանցված թվի ֆակտորիալն է։
- Եթե արգումենտը 0-ից փոքր է, ֆունկցիան վերադարձնում է 0։
- Գերհագեցումները չպետք է մշակվեն, ֆունկցիայի վերադարձը կլինի անորոշ։
- 0-ի 0 աստիճանը պետք է 1 վերադարձնի։
- Նախատիպը պետք է լինի այսպիսին`

int ft_iterative_factorial(int nb);

Գլուխ IV

Առաջադրանք 01 : ft_recursive_factorial

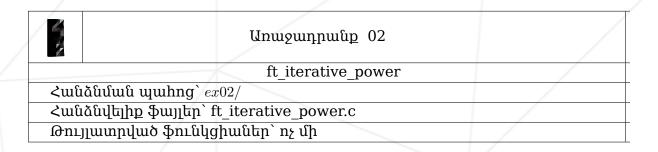


- Ստեղծել ոեկուրսիվ ֆունկցիա, որը վերադարձնում է որպես մուտքային պարամետր փոխանցված թվի ֆակտորիալը։
- Եթե արգումենտր վավեր չէ, ֆունկցիան վերադարձնում է 0։
- Գերհագեցումները չպետք է մշակվեն, ֆունկցիայի վերադարձը կլինի անորոշ։
- Նախատիպը պետք է լինի այսպիսին`

int ft_recursive_factorial(int nb);

Գլուխ V

Առաջադրանք 02 ։ ft_iterative_power

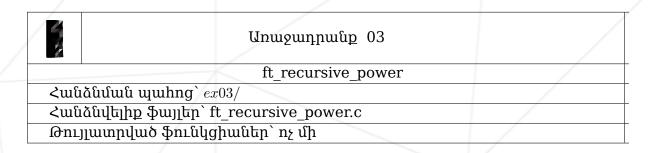


- Ստեղծել իտերացիոն ֆունկցիա, որը վերադարձնում է աստիճան բարձրացրած թվի արժեքը։ 0-ից փոքր աստիճանի դեպքում վերադարձվում է 0։ Գերհագեցումները չպետք է մշակվեն։
- Համարվում է, որ 0-ի 0 աստիճանը 1 է։
- Նախատիպը պետք է լինի այսպիսին`

int ft_iterative_power(int nb, int power);

Գլուխ VI

Unugunpuup 03: ft_recursive_power

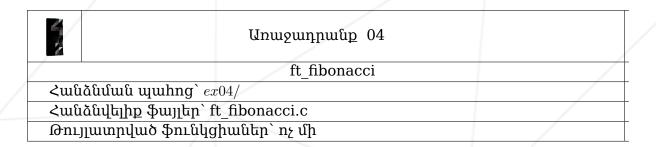


- Ստեղծել ռեկուրսիվ ֆունկցիա, որը վերադարձնում է աստիճան բարձրացրած թվի արժեքը։
- Գերհագեցումները չպետք է մշակվեն, ֆունկցիայի վերադարձը կլինի անորոշ։
- Համարվում է, որ 0-ի 0 աստիճանը 1 է։
- Նախատիպը պետք է լինի այսպիսին`

int ft_recursive_power(int nb, int power);

Գլուխ VII

Առաջադրանը 04 : ft_fibonacci



- Ստեղծել ft_fibonacci ֆունկցիա, որը կվերադարձնի Ֆիբոնաչիի հաջորդականության ո-րդ տարրը։ Առաջին տարրը 0 ինդեքսում է։ Կհամարենք,որ Ֆիբոնաչիի հաջորդականությունը սկսվում է այսպես՝ 0, 1, 1, 2։
- Գերհագեցումները չպետք է մշակվեն, ֆունկցիայի վերադարձը կլինի անորոշ։
- Նախատիպը պետք է լինի այսպիսին`

int ft_fibonacci(int index);

- Բնականաբար, ft_fibonacci-ն պետք է լինի ոեկուրսիվ։
- Եթե index-ը 0-ից փոքր է, ֆունկցիան պետք է վերադարձնի -1։

Գլուխ VIII Առաջադրանք 05 : ft_sqrt

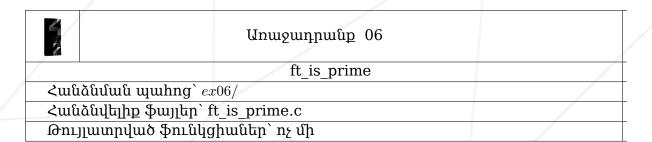
U	ոաջադրանք 05	
/	ft_sqrt	/
Հանձնման պահոց` ex05/	/	
Հանձնվելիք ֆայլեր` ft_sqrt.	/	
Թույլատրված ֆունկցիանե	/	

- Ստեղծել ֆունկցիա, որը վերադարձնում է թվի քառակուսի արմատը (եթե այն գոյություն ունի), կամ 0, եթե թվի արմատն իռացիոնալ թիվ է։
- Նախատիպը պետք է լինի այսպիսին`

int ft_sqrt(int nb);

Գլուխ IX

Առաջադրանք 06 : ft_is_prime



- Ստեղծել ֆունկցիա, որը վերադարձնում է 1, եթե որպես պարամետր փոխանցված թիվը պարզ թիվ է և 0, եթե պարզ չէ։
- Նախատիպը պետք է լինի այսպիսին`

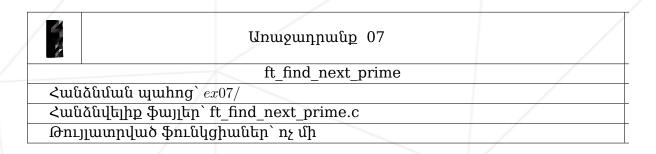
int ft_is_prime(int nb);



0-ն և 1-ը պարզ թվեր չեն։

Գլուխ X

Առաջադրանք 07 : ft_find_next_prime

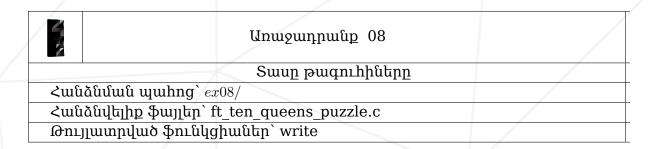


- Ստեղծել ֆունկցիա, որը վերադարձնում է որպես արգումենտ փոխանցված թվից մեծ կամ հավասար հաջորդ պարզ թիվը։
- Նախատիպը պետք է լինի այսպիսին`

int ft_find_next_prime(int nb);

Գլուխ XI

Առաջադրանք 08 ։ Տասը թագուհիները



- Ստեղծել ֆունկցիա, որն արտածում է տասը սյունակ և տասը տող ունեցող շախմատի տախտակի վրա տասը թագուհիների բոլոր հնարավոր դիրքերը` առանց թույլ տալու թագուհիներին իրար հարվածել մեկ քայլով, և վերադարձնում է հնարավոր դիրքերի քանակը։
- Այս խնդիրը լուծելու համար պետք է օգտագործել ռեկուրսիա։
- Նախատիպը պետք է լինի այսպիսին`

```
int ft_ten_queens_puzzle(void);
```

• Այն պետք է արտապատկերվի հետևյալ կերպ`

```
$>./a.out | cat -e
0257948136$
0258693147$
...
4605713829$
4609582731$
...
9742051863$
$>
```

• Հաջորդականությունը ձախից աջ է։ Առաջին թիվը ցույց է տալիս առաջին թագուհու դիրքն առաջին սյունակում (ինդեքսավորումը

սկսվում է 0-ից)։ N-րդ թիվը ցույց է տալիս N-րդ թագուհու դիրքը N-րդ սյունակում։

• Վերադարձվող արժեքը պետք է լինի արտածված լուծումների ընդհանուր քանակը։

Գլուխ XII

<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։



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