



C Ավագան

C 05

Հակիրճ. այս փաստաթուղթը 42 դպրոցի C ավագանի C 05 մողուլի
նյութըն է:

Ցանկ

| | | |
|------|---|----|
| I | Յուցումներ | 2 |
| II | Նախաբան | 4 |
| III | Առաջադրանք 00 : ft_iterative_factorial | 6 |
| IV | Առաջադրանք 01 : ft_recursive_factorial | 7 |
| V | Առաջադրանք 02 : ft_iterative_power | 8 |
| VI | Առաջադրանք 03 : ft_recursive_power | 9 |
| VII | Առաջադրանք 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 թագուհիները | 14 |

Գլուխ I

Ցուցումներ

- Այս էջը ձեր միակ ուղեցույցն է: Պտտվող խոսակցություններին ուշադրություն մի՛ դարձրեք:
- Զգուշացում. մինչ առաջադրանքները հանձնելը նորից ստուգե՛ք նյութը: Ցանկացած պահի այս փաստաթուղթը կարող է փոփոխվել:
- Ուշադրություն դարձրե՛ք ձեր Փայլերի և պահումների թույլտվություններին:
- Բոլոր առաջադրանքները կատարելիս անհրաժեշտ է հետևել հանձման ընթացակարգին:
- Զեր 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, տիաճ 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 Փայլերը 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 Փունկցիաների կիրառումը համարվում է խարդախություն: Խարդախությունը պատժվում է -42-ով, և այս գնահատականը քննարկման ենթակա չէ:

- main () Փունկցիա 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 Փայլից բացի ձեր պահոցում հավելյալ Փայլեր 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 Փորումի «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 Արամագդի: 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 ոեկուրսիան: Ես

նաև գիտեմ, որ դու գիտես, որ ես գիտեմ, որ դու գիտես, որ ես գիտեմ,
որ դու գիտես, ..., թե ինչ է ուկուրսիան:

Գլուխ III

Առաջադրանք 00 : ft_iterative_factorial

| | |
|---|---|
|  | Առաջադրանք 00 ft_iterative_factorial Հանձնման պահng՝ ex00/ Հանձնվելիք Փայլեր՝ ft_iterative_factorial.c Թույլատրված Փունկցիաներ՝ ոչ մի |
|---|---|

- Ստեղծել իտերացիոն Փունկցիա, որը վերադարձնում է թիվ: Այդ թիվը որպես մուտքային պարամետր փոխանցված թվի Փակտորիալն է:
- Եթե արգումենտը վավեր չէ, Փունկցիան վերադարձնում է 0:
- Գերհազեցումները չպետք է մշակվեն, Փունկցիայի վերադարձը կլինի անորոշ:
- Նախատիայր պետք է լինի այսպիսին՝

```
int ft_iterative_factorial(int nb);
```

Գլուխ IV

Առաջադրանք 01 : ft_recursive_factorial

| | |
|---|---|
|  | Առաջադրանք 01 |
| | ft_recursive_factorial |
| | Հանձնման պահng՝ ex01/ |
| | Հանձնվելիք Փայլեր՝ ft_recursive_factorial.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել ոեկուրսիվ Փունկցիա, որը վերադարձնում է որպես մուտքային պարամետր փոխանցված թվի ֆակտորիալը:
- Եթե արգումենտը վավեր չէ, Փունկցիան վերադարձնում է 0:
- Գերհազեցումները չպետք է մշակվեն, Փունկցիայի վերադարձը կլինի անորոշ:
- Նախատիապը պետք է լինի այսպիսին՝

```
int ft_recursive_factorial(int nb);
```

Գլուխ V

Առաջադրանք 02 : ft_iterative_power

| | |
|---|---|
|  | Առաջադրանք 02 |
| | ft_iterative_power |
| | Հանձնման պահng՝ ex02/ |
| | Հանձնվելիք Փայլեր՝ ft_iterative_power.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել իտերացիոն Փունկցիա, 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

Առաջադրանք 03 : ft_recursive_power

| | |
|---|---|
|  | Առաջադրանք 03 |
| | ft_recursive_power |
| | Հանձնման պահng՝ ex03/ |
| | Հանձնվելիք Փայլեր՝ ft_recursive_power.c |
| | Թույլատրված ֆունկցիաներ՝ ոչ մի |

- Ստեղծել ոեկուրսիվ ֆունկցիա, որը վերադարձնում է աստիճան բարձրացրած թվի արժեքը:
- Գերհազեցումները չպետք է մշակվեն, ֆունկցիայի վերադարձը կլինի անորոշ:
- Համարվում է, որ 0-ի 0 աստիճանը 1 է:
- Նախատիապը պետք է լինի այսպիսին՝

```
int ft_recursive_power(int nb, int power);
```

Գլուխ VII

Առաջադրանք 04 : ft_fibonacci

| | |
|---|-----------------------------------|
|  | Առաջադրանք 04 |
| | ft_fibonacci |
| | Հանձնման պահng՝ ex04/ |
| | Հանձնվելիք Փայլեր՝ ft_fibonacci.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել `ft_fibonacci` Փունկցիա, որը կվերադարձնի Ֆիբոնաչիի հաջորդականության ո-րդ տարրը: Առաջին տարրը 0 ինդեքսում է: Կհամարենք,որ Ֆիբոնաչիի հաջորդականությունը սկսվում է այսպէս՝ 0, 1, 1, 2:
- Գերհազեցումները չպետք է մշակվեն, Փունկցիայի վերադարձը կլինի անորոշ:
- Նախատիպը պետք է լինի այսպիսին՝

```
int ft_fibonacci(int index);
```

- Բնականաբար, `ft_fibonacci`-ն պետք է լինի ռեկուրսիվ:
- Եթե `index`-ը 0-ից փոքր է, Փունկցիան պետք է վերադարձնի -1:

Գլուխ VIII

Առաջադրանք 05 : ft_sqrt

| | |
|--|--------------------------------|
| | Առաջադրանք 05 |
| | ft_sqrt |
| | Հանձնման պահng՝ ex05/ |
| | Հանձնվելիք Փայլեր՝ ft_sqrt.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել Փունկցիա, որը վերադարձնում է թվի քառակուսի արմատը (եթե այն գոյություն ունի), կամ 0, եթե թվի արմատն իրացիոնալ թիվ է:
- Նախատիպը պետք է լինի այսպիսին՝

```
int ft_sqrt(int nb);
```

Գլուխ IX

Առաջադրանք 06 : ft_is_prime

| | |
|--|----------------------------------|
| | Առաջադրանք 06 |
| | ft_is_prime |
| | Հանձնման պահng՝ ex06/ |
| | Հանձնվելիք Փայլեր՝ ft_is_prime.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել Փունկցիա, 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

| | |
|---|---|
|  | Առաջադրանք 07 |
| | ft_find_next_prime |
| | Հանձնման պահng՝ ex07/ |
| | Հանձնվելիք Փայլեր՝ ft_find_next_prime.c |
| | Թույլատրված Փունկցիաներ՝ ոչ մի |

- Ստեղծել Փունկցիա, 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

| | |
|--|---|
| | Առաջադրանք 08 |
| | Տասը թագուհիները |
| | Հանձնման պահng՝ ex08/ |
| | Հանձնվելիք Փայլեր՝ ft_ten_queens_puzzle.c |
| | Թույլատրված ֆունկցիաներ՝ write |

- Ստեղծել ֆունկցիա, որն արտածում է տասը այունակ 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(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 $0\text{-}hg$): N -իդ թիվը ցույց է տալիս N -իդ թագուհու դիրքը N -իդ այունակում:

- Վերադարձվող արժեքը պետք է լինի արտածված լուծումների ընդհանուր քանակը: