

C Ավազան C 13

<ակիրճ. այս փաստաթուղթը 42 դպրոցի C ավազանի C 13 մոդուլի նյութն t:

# Ցանկ

•	Singinadaji	
II	Նախաբան	4
III	Առաջադրանք 00 ։ btree_create_node	5
IV	Առաջադրանք 01 ։ btree_apply_prefix	6
V	Առաջադրանք 02 ։ btree_apply_infix	7
VI	Առաջադրանք 03 ։ btree_apply_suffix	8
VII	Առաջադրանք 04 ։ btree_insert_data	9
VIII	Առաջադրանք 05 ։ btree_search_item	10
IX	Առաջադրանք 06 ։ btree_level_count	11
X	Առաջադրանք 07 ։ btree_apply_by_level	12
XI	Հանձնում և ընկերն ընկերոջը ստուգում	13

# 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։

C Ավազան C 13

• 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 էր 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։

:

• <ետագա առաջադրանքների համար կօգտագործենք հետևյալ կառուցվածքը`

- Այս կառուցվածքը պետք է ներառել ft\_list.h ֆայլում և հանձնել այն յուրաքանչյուր առաջադրանքի հետ։
- Հաշվի առե՛ք, որ առաջադրանք 01-ից սկսած՝ օգտագործելու ենք մեր ft\_create\_elem-ը (ցանկալի կլինի, որ ֆունկցիայի նախատիպը լինի ft list.h ֆայլում...)։

#### Գլուխ II

## Նախաբան

SerArt-ը Արտո Թունջբոյաջյանի և Սերժ Թանկյանի համատեղ ալբոմն է, որը թողարկվել է 2003 թվականի մայիսի 6-ին։

Ստորև ալբոմի երգացանկն է, նշված են նաև ստեղծագործության

հեղինակները։

1. «Intro» Թանկյան

2. «Cinema» Թանկյան

3. «Devil's Wedding» Թունջբոյաջյան

4. «The Walking Xperiment» Թանկյան

5. «Black Melon» Թունջբոյաջյան

6. «Metal Shock» Թունջբոյաջյան

7. «Save the Blonde» 🛮 Թանկյան

8. «Love is the Peace» ຜາເນຶ່ງກາງພຽງພົບ

9. «Leave Melody Counting Fear» Թանկյան

10. «Gee-Tar» Թանկյան

11. «Claustrophobia» Թանկյան

12. «Narina» Թունջբոյաջյան, Ձեննա Ռոսս

13. «Zumba» Թունջբոյաջյան

14. «Facing the Plastic» Թանկյան

15. «If You Can Catch Me» ເອກເນອກກາງເພງເພພ

16. «I Don't Want to Go Back Empty-Handed» Թունջբոյաջյան

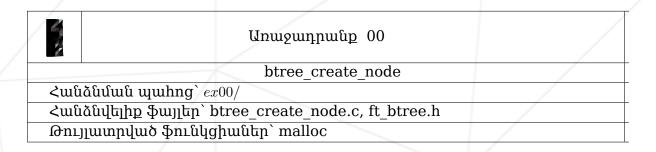
17. «Facing the Plastic» Թանկյան, Mindless Self Indulgence

18. «Narina» Թունջբոյաջյան, Ռոսս

Այսօրվա նյութն ավելի հեշտ կթվա, եթե Serart լսեք։

## Գլուխ III

# Առաջադրանք 00 ։ btree\_create\_node

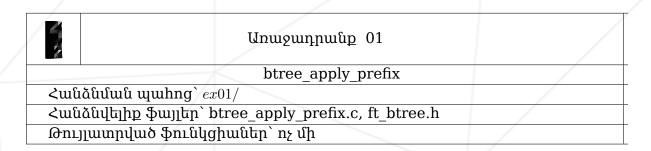


- Ստեղծել btree\_create\_node ֆունկցիան, որը նոր տարր է հատկացնում։ Այն պետք է իր item-ին վերագրի արգումենտի արժեքը, իսկ մնացած տարրերին` 0։
- Վերադարձվում է ստեղծված հանգույցի հասցեն։
- Նախատիպը պետք է լինի այսպիսին`

t\_btree \*btree\_create\_node(void \*item);

### Գլուխ IV

# Առաջադրանք 01 ։ btree\_apply\_prefix

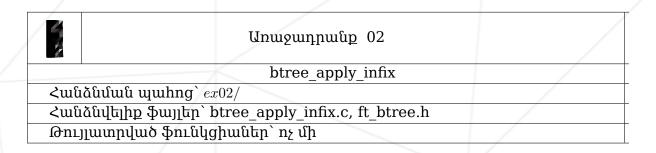


- Ստեղծել btree\_apply\_prefix ֆունկցիա, որը որպես պարամետր փոխանցված ֆունկցիան կիրառում է յուրաքանչյուր հանգույցի item-ին` ծառում փնտրելու համար օգտագործելով prefix տրավերսալը։
- Նախատիպը պետք է լինի այսպիսին`

void btree\_apply\_prefix(t\_btree \*root, void (\*applyf)(void \*));

### Գլուխ V

# Առաջադրանք 02 ։ btree\_apply\_infix

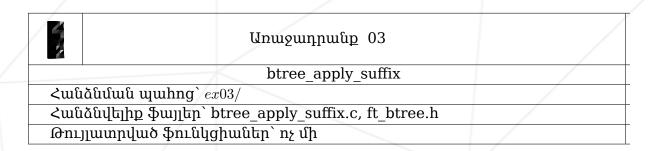


- Ստեղծել btree\_apply\_infix ֆունկցիա, որը որպես պարամետր փոխանցված ֆունկցիան կիրառում է յուրաքանչյուր հանգույցի item-ին` ծառում փնտրելու համար օգտագործելով infix տրավերսալը։
- Նախատիպը պետք է լինի այսպիսին`

void btree\_apply\_infix(t\_btree \*root, void (\*applyf)(void \*));

# Գլուխ VI

# Unwewnpwup 03: btree\_apply\_suffix

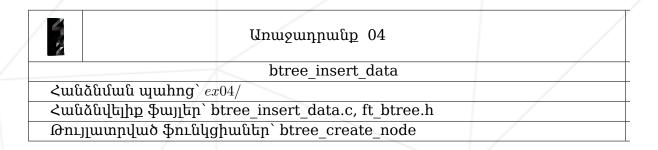


- Ստեղծել btree\_apply\_suffix ֆունկցիա, որը որպես պարամետր փոխանցված ֆունկցիան կիրառում է յուրաքանչյուր հանգույցի item-ին` ծառում փնտրելու համար օգտագործելով suffix տրավերսալը։
- Նախատիպը պետք է լինի այսպիսին`

void btree\_apply\_suffix(t\_btree \*root, void (\*applyf)(void \*));

### Գլուխ VII

# Առաջադրանք 04 ։ btree\_insert\_data

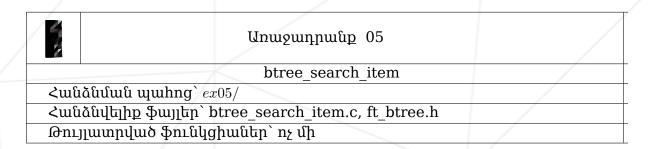


- Ստեղծել btree\_insert\_data ֆունկցիա, որը տեղադրում է item տարրը ծառի մեջ։ Որպես արգումենտ փոխանցված ծառը պետք է դասավորվի այպես. յուրաքանչյուր հանգույցի համար բոլոր ցածր տարրերը գտնվում են ձախում, իսկ բոլոր բարձր կամ հավասար տարրերը` աջում։ Որպես արգումենտ նաև կփոխանցվի strcmp-ի տիպի համեմատության ֆունկցիա։
- root պարամետրը ծառի արմատային հանգույցի ցուցիչն է։ Առաջին անգամ կանչելիս, այն պետք է ցույց տա NULL-ին։
- Նախատիպը պետք է լինի այսպիսին`

void btree\_insert\_data(t\_btree \*\*root, void \*item, int (\*cmpf)(void \*, void \*));

### Գլուխ VIII

# Առաջադրանք 05 ։ btree\_search\_item

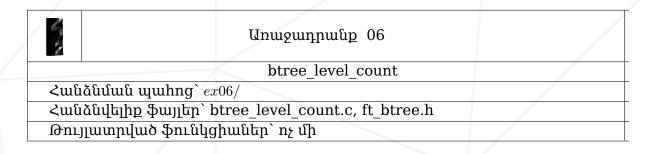


- Ստեղծել btree\_search\_item ֆունկցիա, որը վերադարձնում է որպես արգումենտ փոխանցված հղման տվյալներին համապատասխանող առաջին տարրը։ Ծառը պետք է զննվի infix տրավերսալով։ Եթե տարրը չի գտնվում, ֆունկցիան պետք է վերադարձնի NULL:
- Նախատիպը պետք է լինի այսպիսին`

void \*btree\_search\_item(t\_btree \*root, void \*data\_ref, int (\*cmpf)(void \*, void \*));

# Գլուխ IX

# Առաջադրանք 06 : btree\_level\_count

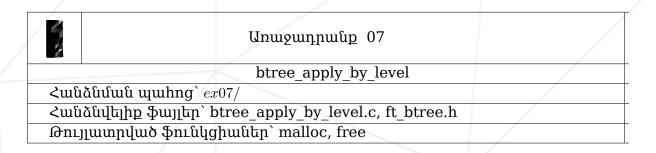


- Ստեղծել btree\_level\_count ֆունկցիա, որը վերադարձնում է որպես արգումենտ փոխանցված ամենամեծ ճյուղի չափը։
- Նախատիպը պետք է լինի այսպիսին`

int btree\_level\_count(t\_btree \*root);

### Գլուխ X

# Unաջադրանք 07 : btree\_apply\_by\_level



- Ստեղծել btree\_apply\_by\_level ֆունկցիա, որը որպես արգումենտ փոխանցված ֆունկցիան կիրառում է ծառի յուրաքանչյուր հանգույցի վրա։ Ծառը պետք է զննվի մակարդակ առ մակարդակ։ Կանչված ֆունկցիան պետք է ընդունի երեք պարամետը.
  - o Առաջին պարամետրը՝ void \* տիպի, համապատասխանում է հանգույցի item-ին։
  - Երկրորդ պարամետրը` int տիպի, համապատասխանում է այն մակարդակին, որի վրա գտնվում է. 0` արմատի համար, 1` երեխաների համար, 2` թոռների համար, և այլն։
  - o Երրորդ int տիպի պարամետրի արժեքը 1 է, եթե այն մակարդակի առաջին հանգույցն է, կամ 0` այլ դեպքերում։
- Նախատիպը պետք է լինի այսպիսին`

void btree\_apply\_by\_level(t\_btree \*root, void (\*applyf)(void \*item, int current\_level, int is\_first

## Գլուխ XI

# <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։