



C Ավագան

C 13

Հակիրճ. այս փաստաթուղթը 42 դպրոցի C ավագանի C 13 մոդուլի նյութն է:

Ցանկ

I	Ցուցումներ	2
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

Ցուցումներ

- 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:
- Բացի դրանից, ձեր առաջադրանքները կստուգվեն և կգնահատվեն Moulinette կոչվող ծրագրով:
- Moulinette-ը գնահատելիս շատ բժախնդիր 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:
- :
- Հետագա առաջադրանքների համար կօգտագործենք հետևյալ կառուցվածքը՝

```
typedef struct      s_btree
{
    struct s_btree  *left;
    struct s_btree  *right;
    void            *item;
}                   t_btree;
```

- 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 թունջբոյաջանի 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» Թունջբոյաջան
4. «The Walking Xperiment» Թանկյան
5. «Black Melon» Թունջբոյաջան
6. «Metal Shock» Թունջբոյաջան
7. «Save the Blonde» Թանկյան
8. «Love is the Peace» Թունջբոյաջան
9. «Leave Melody Counting Fear» Թանկյան
10. «Gee-Tar» Թանկյան
11. «Claustrophobia» Թանկյան
12. «Narina» Թունջբոյաջան, Ջեննա Ռոսս
13. «Zumba» Թունջբոյաջան
14. «Facing the Plastic» Թանկյան
15. «If You Can Catch Me» Թունջբոյաջան
16. «I Don't Want to Go Back Empty-Handed» Թունջբոյաջան
17. «Facing the Plastic» Թանկյան, Mindless Self Indulgence
18. «Narina» Թունջբոյաջան, Ռոսս

Այսօրվա նյութն ավելի հեշտ կթվա, եթե [Serart](#) լսեք:

Գլուխ III

Առաջադրանք 00 : btree_create_node


	Առաջադրանք 00
	btree_create_node
	Հանձնման պահոց՝ <i>ex00/</i>
	Հանձնվելիք ֆայլեր՝ btree_create_node.c, ft_btree.h
	Թույլատրված ֆունկցիաներ՝ malloc

- 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


	Առաջադրանք 01
	btree_apply_prefix
	Հանձնման պահոց՝ <i>ex01/</i>
	Հանձնվելիք ֆայլեր՝ <i>btree_apply_prefix.c, ft_btree.h</i>
	Թույլատրված ֆունկցիաներ՝ <i>ոչ մի</i>

- 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


	Առաջադրանք 02
	btree_apply_infix
	Հանձնման պահոց՝ <i>ex02/</i>
	Հանձնվելիք ֆայլեր՝ <i>btree_apply_infix.c, ft_btree.h</i>
	Թույլատրված ֆունկցիաներ՝ <i>ոչ մի</i>

- 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

Առաջադրանք 03 : btree_apply_suffix


	Առաջադրանք 03
btree_apply_suffix	
Հանձնման պահոց՝ <i>ex03/</i>	
Հանձնվելիք ֆայլեր՝ <i>btree_apply_suffix.c, ft_btree.h</i>	
Թույլատրված ֆունկցիաներ՝ <i>նչ մի</i>	

- 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


	Առաջադրանք 04
	btree_insert_data
	Հանձնման պահոց՝ ex04/
	Հանձնվելիք ֆայլեր՝ btree_insert_data.c, ft_btree.h
	Թույլատրված ֆունկցիաներ՝ btree_create_node

- 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


	Առաջադրանք 05
	btree_search_item
	Հանձնման պահոց՝ <i>ex05/</i>
	Հանձնվելիք ֆայլեր՝ btree_search_item.c, ft_btree.h
	Թույլատրված ֆունկցիաներ՝ ոչ մի

- 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


	Առաջադրանք 06
btree_level_count	
Հանձնման պահոց՝ <i>ex06/</i>	
Հանձնվելիք ֆայլեր՝ btree_level_count.c, ft_btree.h	
Թույլատրված ֆունկցիաներ՝ ոչ մի	

- 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

Առաջադրանք 07 : btree_apply_by_level

	Առաջադրանք 07
	btree_apply_by_level
	Հանձնման պահոց՝ ex07/
	Հանձնվելիք ֆայլեր՝ btree_apply_by_level.c, ft_btree.h
	Թույլատրված ֆունկցիաներ՝ malloc, free

- Ստեղծել btree_apply_by_level ֆունկցիա, որը որպես արգումենտ փոխանցված ֆունկցիան կիրառում է ծառի յուրաքանչյուր հանգույցի վրա: Ծառը պետք է գննվի մակարդակ առ մակարդակ: Կանչված ֆունկցիան պետք է ընդունի երեք պարամետր.
 - 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:
 - 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(elem)))
```

Գլուխ XI

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: