



C Ավագան

C 08

Հակիրճ. այս փաստաթուղթը 42 դպրոցի C ավագանի C 08 մողուլի
նյութը է:

Յանկ

I	Յուցումներ	2
II	Նախաբան	4
III	Առաջադրանք 00 : ft.h	5
IV	Առաջադրանք 01 : ft_boolean.h	6
V	Առաջադրանք 02 : ft_abs.h	8
VI	Առաջադրանք 03 : ft_point.h	9
VII	Առաջադրանք 04 : ft_strs_to_tab	10
VIII	Առաջադրանք 05 : ft_show_tab	12

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:

Գլուխ II

Նախաբան

Ահա թե ինչ է ասում *Վիրիպեղիան բաղակտուցի* մասին:

Բաղակտուցը (լատ.՝ *Ornithorhynchus anatinus*) երկենցաղ, ձվադրող կաթնասուն է՝ ենդեմիկ արևելյան Ավստրալիայի ինչպես նաև Տասմանիայի բնաշխարհում։ Եքիդնայի չորս տեսակների հետ միասին այն միանցքավորների կարգի հինգ վերացող տեսակներից մեկն է, միակ կաթնասունը, որ ձվադրում է, այլ ոչ թե կենդանի ձագեր ծնում։ Չնայած հայտնաբերվել են որոշ ազգակից տեսակների բրածոներ՝ բաղակտուցը իր ընտանիքի (Ornithorhynchidae) ու գենի (Ornithorhynchus) միակ ներկայացուցիչն է։

Այս ձվադրող, բաղի կտուցով, կուղրի պոչով, ջրասամույրի ոտքերով կաթնասունի տարօրինակ տեսքը շփոթեցրել եր եվրոպացի բնագետներին, երբ վերջիններս առաջին անգամ նկատել եին կենդանուն։ Մի մասը նույնիսկ այն կարծիքին եր, որ իրենց աչքերի առաջ լավ մտածված խաբեություն է։ Բաղակտուցը պատկանում է այն փոքրաթիվ կաթնասունների շարքին, որոնք թունավոր են։ Արու բաղակտուցը ոտքերի հետևում ճանկ ունի, որի թույնը խիստ ցավ կարող է պատճառել մարդկանց։ Իր յուրօրինակ հատկանիշների շնորհիվ բաղակտուցը մեծ դեր ունի եվոյցիցին կենսաբանության մեջ և համարվում է Ավստրալիայի խորհրդանիշը։ Այն թայխամանի դեր ունի ազգային միջոցառումներում և պատկերված է 20 ցենտանոցի հակառակ կողմում։ Բաղակտուցը նաև Նոր Հարավային Ուելսի խորհրդանիշ-կենդանին է։

Մինչև 20-րդ դարի սկզբները կենդանուն որսում եին իր մորթու համար, սակայն ներկայումս բոլոր տեսակները պաշտպանված են։ Չնայած արհեստական բազմացման ծրագրերը շատ սահմանափակ հաջողություն են ունեցել, և բաղակտուցը պաշտպանված չէ օդի աղտոտման վնասներից՝ ներկայումս կենդանին ոչ մի անմիջական վտանգի տակ չէ։

Այս թեման ընդհանրապես կապ չունի բաղակտուցների հետ։

Գլուխ III

Առաջադրանք 00 : ft.h

	Առաջադրանք 00
	ft.h
Հանձնման պահNG՝ ex00/	
Հանձնվելիք Փայլեր՝ ft.h	
Թույլատրված Փունկցիաներ՝ ոչ մի	

- Ստեղծել ft.h Փայլ:
- Այն պարունակում է բոլոր Փունկցիաների նախատիպերը՝

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

Գլուխ IV

Առաջադրանք 01 : ft_boolean.h

	Առաջադրանք 01
	ft_boolean.h
	Հանձնման պահng՝ ex01/
	Հանձնվելիք Փայլեր՝ ft_boolean.h
	Թույլատրված Փունկցիաներ՝ nշ մի

- Ստեղծել `ft_boolean.h` Փայլ: Այն կկազմարկի և կաշիատեցնի հետևյալ `main`-ը համաստասիանաբար՝

```
#include "ft_boolean.h"

void          ft_putstr(char *str)
{
    while (*str)
        write(1, str++, 1);
}

t_bool        ft_is_even(int nbr)
{
    return ((EVEN(nbr)) ? TRUE : FALSE);
}

int           main(int argc, char **argv)
{
    (void)argv;
    if (ft_is_even(argc - 1) == TRUE)
        ft_putstr(EVEN_MSG);
    else
        ft_putstr(ODD_MSG);
    return (SUCCESS);
}
```

- Ծրագիրը պետք է արտածի՝

```
%  
I have an even number of arguments.
```

- կամ

```
%  
I have an odd number of arguments.
```

- որին հարկ եղած դեպքում հետևում է դատարկ տող:



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

Գլուխ V

Առաջադրանք 02 : ft_abs.h

	Առաջադրանք 02
	ft_abs.h
	Հանձնման պահng` ex02/
	Հանձնվելիք Փայլեր՝ ft_abs.h
	Թույլատրված Փունկցիաներ՝ ոչ մի

- Ստեղծել մակրո ABS, որը իր արգումենտը փոխարինում է դրա բացարձակ արժեքով:

```
#define ABS(Value)
```



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

Գլուխ VI

Առաջադրանք 03 : ft_point.h

	Առաջադրանք 03
	ft_point.h
	<անձնման պահng` ex03/
	<անձնվելիք Փայլեր` ft_point.h
	Թույլատրված Փունկցիաներ՝ ոչ սի

- Ստեղծել `ft_point.h` Փայլ, որը կկազմարկի հետևյալ `main`-ը՝

```
#include "ft_point.h"

void      set_point(t_point *point)
{
    point->x = 42;
    point->y = 21;
}

int      main(void)
{
    t_point      point;

    set_point(&point);
    return (0);
}
```

Գլուխ VII

Առաջադրանք 04 : ft_strs_to_tab

	Առաջադրանք 04
	ft_strs_to_tab
	Հանձնման պահng՝ ex04/
	Հանձնվելիք Փայլեր՝ ft_strs_to_tab.c
	Թույլատրված ֆունկցիաներ՝ malloc, free

- Ստեղծել ֆունկցիա, որը որպես արգումենտ ստանում է տողերի զանգված և այդ զանգվածի չափը:
- Նախատիպը պետք է լինի այսպիսին՝

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- Այն փոխարինում է ac-ի յուրաքանչյուր տարրը ստրուկտուրայով:
- Այդ ստրուկտուրան սահմանված կլինի `ft_stock_str.h` Փայլում հետևյալ կերպ՝

```
typedef struct s_stock_str
{
    int size;
    char *str;
    char *copy;
} t_stock_str;
```

- size-ը տողի երկարությունն է,
- str-ն տողն է,
- copy-ն տողի կրկնօրինակն է:

- Պետք է պահպանվի առ-ի հաջորդականությունը:
- Վերադարձված զանգվածը պետք է տեղակայվի հիշողության մեջ, և դրա վերջին տարրի str-ն 0 է, որը նշում է զանգվածի վերջը:
- Սխալի դեպքում NULL ցուցիչ պետք է վերադարձվի:
- Մենք կփորձարկենք ձեր ֆունկցիան մեր ft_show_tab-ով (հաջորդ վարժություն): Ստուգե՛ք ֆունկցիայի աշխատանքը համապատասխանաբար:

Գլուխ VIII

Առաջադրանք 05 : ft_show_tab

	Առաջադրանք 05
	ft_show_tab
	<անձնման պահng` ex05/
	<անձնվելիք Փայլեր՝ ft_show_tab.c
	Թույլատրված Փունկցիաներ՝ write

- Ստեղծել Փունկցիա, որն արտածում է նախորդ Փունկցիայով կառուցված զանգվածի տարրերը:
- Նախատիպը պետք է լինի այսպիսին՝

```
void ft_show_tab(struct s_stock_str *par);
```

- Ստրուկտուրան պետք է նախորդ վարժության նման լինի և կսահմանվի `ft_stock_str.h` Փայլում:
- Յուրաքանչյուր տարրի համար կարտածենք՝
 - տողը, որին պետք է հաջորդի '\n'
 - չափը, որին պետք է հաջորդի '\n'
 - տողի կրկնօրինակը, (որը կարող է ձևափոխված լինել) որին պետք է հաջորդի '\n'
- Զեր Փունկցիան կփորձարկենք մեր `ft_strs_to_tab`-ով (նախորդ առաջադրանք): Ստուգե՛ք Փունկցիայի աշխատանքը համապատասխանաբար: