



# C Ավագան

## Rush 02

*Հակիրճ. այս փաստաթուղթը 42 դպրոցի C ավագանի Rush02 մոդուլի նյութն է:*

# Ցանկ

<b>I</b>	<b>Ցուցումներ</b>	<b>2</b>
<b>II</b>	<b>Նախաբան</b>	<b>4</b>
<b>III</b>	<b>Նյութ</b>	<b>6</b>
<b>IV</b>	<b>Բոնուս</b>	<b>8</b>
<b>V</b>	<b>Հանձնում և ընկերն ընկերոջը ստուգում</b>	<b>9</b>

# Գլուխ I

## Ցուցումներ

- Խուսմբը պաշտպանությանը կգրանցվի ավտոմատ կերպով:
- Մի չեղարկեք այն, երկրորդ շանս չի տրվելու.:
- Նյութին վերաբերվող ցանկացած հարց կրարդացնի նյութը:
- Բոլոր առաջադրանքների համար պետք է հետևել հանձնման ընթացակարգին:
- Այս նյութը կարող է փոխվել հանձնումից մինչև մեկ ժամ առաջ:
- Ձեր ծրագիրը պետք է կազմարկվի -Wall -Wextra -Werror դրոշակների օգնությամբ, գործածելով cc:
- Եթե ձեր ծրագիրը չկազմարկվի՝ կգնահատվեք 0:
- Ձեր ծրագիրը պետք է գրված լինի Norm-ին համապատասխան: Եթե ունեք լրացուցիչ ֆայլեր կամ ֆունկցիաներ՝ դրանց համապատասխանությունը Norm-ին ևս ստուգվում է, և Norm-ի սխալի դեպքում կգնահատվեք 0:
- Պետք է հանձնել Makefile, որը կկազմարկի ծրագիրը՝ օգտագործելով \$NAME, clean և fclean կանոնները:
- Այսպիսով, նախագիծը պետք է կատարել ձեզ կցված թիմի հետ և պաշտպանության ժամին ներկայանալ ձեր բոլոր թիմակիցների հետ:
- Աշխատանքը պետք է ավարտված լինի մինչ պաշտպանության օրը: Պաշտպանության նպատակը աշխատանքի մանրամասները ներկայացնելն ու պարզաբանելն է:
- Թիմի բոլոր անդամները պետք է աշխատանքի բոլոր մանրամասներին քաջատեղյակ լինեն: Եթե աշխատանքի բաժանում է կատարվել, պետք է այնպես անել, որ թիմի յուրաքանչյուր անդամ լիովին տեղյակ լինի մյուս անդամների աշխատանքի բոլոր նրբություններին: Պաշտպանության ընթացքում ձեզ տրվելու են հարցեր, և վերջնական գնահատականը հիմնված է լինելու վատագույն պատասխանների վրա:

- Թիմին հավաքելը ձեր պարտականությունն է: Հաղորդակցման ցանկացած միջոց ձեր տրամադրության տակ է՝ հեռախոս, էլեկտրոնային փոստ, փոստատար աղավնի, ոգեղենություն և այլն: Այնպես որ իզուր պատրվակներ չբստրե՛ք: Կյանքը միշտ չէ, որ արդար է, ուզեք թե չուզեք:
- Այնուամենայնիվ, եթե իսկապես ամեն բան փորձել եք, բայց ձեր թիմակիցներից մեկը շարունակում է անհասանելի լինել, միևնույն է, կատարե՛ք աշխատանքը. պաշտպանության ժամանակ կփորձենք տեսնել, թե ինչ կարելի է անել: Նույնիսկ եթե խմբի ղեկավարը բացակա է, հանձնման պահոցը միևնույն է ձեզ համար հասանելի կլինի:

# Գլուխ II

## Նախաբան

Ահա փախլավայի պատրաստման մի լավ բաղադրատոմս.

Բաղադրիչներ՝  
Խմորի համար՝  
ձու - 1 հատ  
կարագ - 75գր  
կաթ - 60մլ  
այլուր - 2,5 բաժակ  
շաքարավազ - 1/2թ/գդ  
փխրեցուցիչ - 1/2թ/գդ  
աղ - 1/2թ/գդ

Միջուկի համար՝  
ընկույզ - 300գր  
շաքարավազ - 150գր  
կոճապղպեղ (имбирь) - 1թ/գդ  
դարչին - 1/2թ/գդ

Վրան լցնելու համար՝  
կարագ - 100գր  
ձու - 1 հատ  
շաքարավազ - 250գր  
ջուր - 1/4 բաժակ  
մեղր - 3 ճ/գդ

Պատրաստման եղանակը՝

Խմորի համար տաքացնում ենք կաթը, լցնում բաժակի մեջ, ավելացնում ենք շաքարավազն ու փխրեցուցիչը, խառնում և թողնում հանգստանա 10-15 րոպե: Խառնում ենք աղը, այլուրը, ավելացնում կարագը, ձուն, շաքարավազով ու փխրեցուցիչով խառնած կաթը և հունցում փափուկ, հարթ խմոր (այնպես, որ համարյա չկաշի ձեռքերին): Թողնում ենք խմորը հանգստանա 1 ժամ տաք վայրում:  
Միջուկի համար մանր կտրատում ենք ընկույզը, խառնում շաքարավազի, կոճապղպեղի, դարչինի հետ:

Ապա ձեռքերով ճմրթում ենք խմորը, թաթախում այլուրի մեջ և բաժանում 9 գնդի (մի գունդը անում ենք մի փոքր ավելի մեծ): Գրտնակում ենք խմորը, բացում բարակ 1 մմ շերտով և դնում նախապես կարագով պատած ձևամանի մեջ, վրան քսում ենք հալեցրած կարագը: Բացում ենք ևս մեկ շերտ խմոր և դնում առաջին շերտի վրա (2 շերտ խմոր իրար վրա անհրաժեշտ է միայն ներքևի շերտում), վրան լցնում ենք 3-46/գդ միջուկ:

Կրկնում ենք նույն գործողությունը մինչև մնա խմորի վերջին մեծ գունդը:

Բացում ենք խմորի մեծ գունդը 2-3 մմ շերտով և փռում փախլավայի վրա՝ ծածկելով այն, վրան քսում ենք հարած ձուն և կտրատում ռոմբերով (կտրատել չհասնելով մինչև ներքևի շերտը)՝ ձև տալով փախլավային: Ամեն ռոմբի մեջտեղում դնում ենք կես ընկույզ և թխում ջեռոցում 10 րոպե 180 աստիճանում:

Ապա հանում ենք փախլավան, նորից դանակով անցնում կտրատված տեղերով, այդ արանքների մեջ լցնում ենք հալեցրած կարագը ու կրկին դնում ենք ջեռոցի մեջ և թխում ևս 20 րոպե:

Սիրուպի համար խառնում ենք շաքարավազը, ջուրը, մեղրը և եփում թույլ կրակի վրա 10 րոպե՝ անընդհատ խառնելով:


Հանում ենք ջեռոցից պատրաստի փախլավան կրկին դանակով անցնում կտրատված տեղերով, վրան լցնում ենք քաղցր սիրուպը, թողնում ենք 1 ժամ, որ սառչի և կարելի է մատուցել:

Մթերքի այս քանակությունից ստացվում է 6 բաժին փախլավա: Եթե ընկույզի սիրահար չեք, կարելի է փոխարինել ընկույզը պնդուկով :

Նկարները կարող եք տեսնել [այստեղ](#):

# Գլուխ III

## Նյութ

	Առաջադրանք 00
	rush-02
	Հանձնման պահոց՝ <i>ex00/</i>
	Հանձնվելիք ֆայլեր՝ Makefile և բոլոր անհրաժեշտ ֆայլեր
	Թույլատրված ֆունկցիաներ՝ write, malloc, free, open, read, close

- Ստեղծել ծրագիր, որը որպես արգումենտ փոխանցված թիվը վերափոխում է իր բառային համարժեքին:
- Գործարկվող ծրագրի անունը՝ *rush-02*:
- Մուտքային կոդը կկազմարկվի այսպես՝

```
make fclean
make
```

- Ծրագիրը կարող է ընդունել առավելագույնը երկու արգումենտներ:
  - Միայն մեկ արգումենտի դեպքում դա այն արժեքն է, որը պետք է վերափոխել:
  - Երկու արգումենտի դեպքում առաջին արգումենտը հղման նոր բառարանն է, իսկ երկրորդը այն արժեքն է, որը պետք է վերափոխել:
- Եթե արգումենտը վավեր ամբողջ թիվ չէ, ծրագիրը պետք է վերադարձնի "Error" հաջորդված նոր տողով:



Ձեր ծրագիրը պետք է գերազանցի առանց նշանի *int*-ը:

- Ծրագիրը պետք է վերծանի որպես աղբյուր տրված բառարանը: Բառարանի արժեքները պետք է օգտագործվեն արդյունքի արտատպման համար: Այս արժեքները կարող են փոխվել:
- malloc(3)-ով դինամիկ հիշողության մեջ ցանկացած բաշխում պետք է ճիշտ ազատվի: Սա կստուգվի գնահատման ժամանակ:
- Բառարանը կունենա հետևյալ կանոնները`

```
[a number][0 to n spaces]:[0 to n spaces][any printable characters]\n
```

- Թվերը պետք է փոխակերպվեն այնպես, ինչպես atoi-ն է անում:
- Բառարանի արժեքներին նախորդող և հաջորդող բացատները պետք է կտրվեն:
- Բառարանը պետք է պարունակի առնվազն այն բանալիները , որոնք կան հղման բառարանում: Դրանց արժեքը կարող է փոփոխվել, կարող են լինել ավելացումներ, բայց սկզբնական բանալիները չեն կարող հեռացվել:
- Պետք է միայն օգտագործել սկզբնական գրառումները (օրինակ՝ 54: fifty-four ավելացնելու դեպքում, միևնույն է պետք է օգտագործել 50: fifty և 4: four):
- Բառարանի գրառումները կարող են պահվել ցանկացած դասավորությամբ:
- Բառարանում կարող են լինել դատարկ տողեր:
- Եթե կան բառարանի վերծանման սխալներ, ծրագիրը պետք է տպի "Dict Error\n":
- Եթե անհրաժեշտ արժեքը բառարանում չի գտնվում, ծրագիրը պետք է տպի Error\n".
- Օրինակ՝

```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
$> grep "20" numbers.dict | cat -e
20 :      hey      everybody !$
$> ./rush-02 20 | cat -e
hey everybody !$
```



## Գլուխ IV

### Բոնուս

- Օգտագործե՛ք -, ,, and` գրավոր ճիշտ շարադասությանը ավելի նման մնալու համար:
- Նույն առաջադրանքը այլ լեզուներով կատարելիս թույլատրվում է օգտագործել այլ բառարան, որը կպարունակի բոլոր անհրաժեշտ գրառումները:
- Օգտագործել read` արգումենտ չունենալու դեպքում ստանդարտ գրառում կարդալու համար

## Գլուխ V

# 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:

Քանի որ այս առաջադրանքները չեն ստուգվում որևէ ծրագրի կողմից՝ ազատորեն կարող եք օպտիմալացնել ձեր ֆայլերն ինչպես ցանկանում եք, քանի դեռ հանձնում եք պարտադիր ֆայլերը և բավարարում նախագծի պահանջները:



Հարկավոր է հանձնել միայն այն ֆայլերը, որոնք պահանջվում են այս նախագիծը նկարագրող ֆայլում: