

Խեցգետին-գրաշար

Որոշ մարդիկ ասում են, որ Լեոնարդոյն Յոհանես Գոտեմբերգի՝ գերմանացի դարբինի, մեծ երկրպագուն էր, ով հայտնագործել էր շարժական տպագրական մեքենան, և ի պատիվ նրա նախագծեց խեցգետին-գրաշար — *il gambero scrivano* կոչվող մեքենան՝ շատ հասարակ տպագրական սարք։ Այն շատ մեծ է ժամանակակից տպագրական մեքենաներին և ընդունում է միայն երկու հրաման. մեկը տպել հաջորդ սիմվոլը, մյուսը՝ չեղյալ համարել ամենավերջին հրամանները։ Խեցգետին-գրաշարի մշակողի հատկությունն այն է, որ `undo` հրամանը չափազանց հզոր է. `undo`-ն նույնպես հրաման է և նույնպես կարող է չեղյալ համարվել։

Խնդիրը

Ձեր խնդիրն է գրել խեցգետին-գրաշարի ծրագրային տարբերակը։ Այն սկսում է աշխատանքը դատարկ տեքստից և ընդունում է օգտվողի կողմից տրված հրամանների հաջորդականություն, և հարցումներ տեքստի ընթացիկ տարբերակում որոշակի դիրքերի վերաբերյալ, ինչպես նկարագրված է ստորև։

- `Init()` — կանչվում է մեկ անգամ ամենակզբում, առանց պարամետրերի։ Այն կարող է օգտագործել տվյալների կառուցվածքները սկզբնաթմբավորելու համար։ Այս հրամանը `undo` չի արվում։
- `TypeLetter(L)` — տեքստի վերջում ավելացնում է `L` փոփոխատուը `a, ..., z` տիրույթից։
- `UndoCommands(U)` — վերջին `U` հրամանները չեղյալ է դարձնում, `U`-ն դրական ամբողջ թիվ է։
- `GetLetter(P)` — վերադարձնում է ընթացիկ տեքստի `P` դիրքում գտնվող տառը, `P` ոչ բացասական ինդեքսի համար։ Տեքստում առաջին տառի ինդեքսը 0 է։ (Այս հարցումը հրաման չէ, և, հետևաբար, անտեսվում է `undo` հրամանի կողմից)։

Սկզբնական `Init()` կանչից հետո մնացած եզրածրագրերը կարող են կանչվել զրո կամ ավել անգամ կամայական հերթականությամբ։ Երաշխավորվում է, որ `U`-ն չի գերազանցի մինչև այդ կատարված հրամանների քանակը, և, որ `P`-ն փոփոխվելու ընթացիկ տեքստի երկարությունից (ընթացիկ տեքստում տառերի քանակը)։

`UndoCommands(U)` հրամանը վերջին `U` հրամանները չեղյալ է անում *հակառակ* կարգով։ Եթե չեղյալ արվող հրամանը `TypeLetter(L)` տիպի է, ապա այն հանում է `L`-ը ընթացիկ տեքստի վերջից։ Եթե չեղյալ արվող հրամանը `UndoCommands(X)` տիպի է,

այն վերա-կատարում է վերջին `X` հրամանները զրանց "ճախակալ" կարգով։

Օրինակ

Ստորև բերված է կանչերի հաջորդականություն, յուրաքանչյուր կանչից հետո տեքստի վիճակով հանդերձ։

Կանչ	Վերադարձնում է	Ընթացիկ տեքստ
Init()		
TypeLetter(a)		a
TypeLetter(b)		ab
GetLetter(1)	b	ab
TypeLetter(d)		abd
UndoCommands(2)		a
UndoCommands(1)		abd
GetLetter(2)	d	abd
TypeLetter(e)		abde
UndoCommands(1)		abd
UndoCommands(5)		ab
TypeLetter(c)		abc
GetLetter(2)	c	abc
UndoCommands(2)		abd
GetLetter(2)	d	abd

Ենթախնդիր 1 [5 միավոր]

- Նրա մանրացման և հարցումների ընդհանուր քանակը 1-ից 100 (ներառյալ) սահմաններում է, և UndoCommands կանչ չկա:

Ենթախնդիր 2 [7 միավոր]

- Նրա մանրացման և հարցումների ընդհանուր քանակը 1-ից 100 (ներառյալ) սահմաններում է, և ոչ մի UndoCommands հրաման չեղյալ չի արվի:

Ենթախնդիր 3 [22 միավոր]

Նրա մանրացման և հարցումների ընդհանուր քանակը 1-ից 5 000 սահմաններում է (ներառյալ):

Ենթախնդիր 4 [26 միավոր]

- Նրա մանրացման և հարցումների ընդհանուր քանակը 1-ից 1 000 000 (ներառյալ) սահմաններում է: Բոլոր GetLetter կանչերը կհանդիպեն թույլ TypeLetter և UndoCommands կանչերից հետո:

Ենթախնդիր 5 [40 միավոր]

Նրա մանրացման և հարցումների ընդհանուր քանակը 1-ից 1 000 000 սահմաններում է (ներառյալ):

Իրականացման մանրամասներ

Դուք պետք է submit անեք ձեր լուծումը, որի անունը պետք է լինի scrivener.c, scrivener.cpp կամ scrivener.pas: Այդ լուծումը պետք է իրականացնի վերը նկարագրված ենթախնդիրները օգտագործելով հետևյալ սիմվոլները:

C/C++ ծրագրեր

```
void Init();
void TypeLetter(char L);
void UndoCommands(int U);
char GetLetter(int P);
```

Pascal ծրագրեր

```
procedure Init;  
procedure TypeLetter(L : Char);  
procedure UndoCommands(U : LongInt);  
function GetLetter(P : LongInt) : Char;
```

Այս եմթածրագրերը պետք է աշխատեն այնպես, ինչպես նկարագրված է վերևում: Իհարկե դուք կարող եք իրականացնել այլ եմթածրագրեր ևս՝ նրանց ներքին օգտագործման համար: Ձեր submit-ները չպետք է գործ ունենան ոչ ստանդարտ մուտք/ելքի հետ, ոչ որևէ այլ ֆայլի հետ:

grader-ի օրինակ

Grader-ի օրինակը կարդում է մուտքը հետևյալ ֆորմատով.

- line 1: մուտքում հրամանների և հարցումների ընդհանուր քանակը.
 - T ապա պրոբել և մի փոքրատառ TypeLetter հրամանի համար;
 - U ապա պրոբել և մի ամբողջ թիվ UndoCommands հրամանի համար;
 - P ապա պրոբել և մի ամբողջ թիվ GetLetter հրամանի համար:

Grader-ի օրինակը տպում է GetLetter-ի վերադարձրած սիմվոլները, յուրաքանչյուրն առանձին տողում: