

მწერალი კიბორჩხალა

ზოგიერთი მკვლევარი თვლის, რომ ლეონარდო იყო გერმანელი მჭედლის
იოჰან გუტენბერგის დიდი თაყვანისმცემელი, რომელმაც გამოიგონა
საბეჭდი მანქანა და მის საპატივცემულოდ შექმნა *il gambero scrivano* —
მწერალი კიბორჩხალა, რომელიც წარმოადგენს მარტივ საბეჭდ
მოწყობილობას. ის ნაწილობრივ წააგავს თანამედროვე საბეჭდ მანქანას და
ასრულებს მხოლოდ ორ ბრძანებას: ერთი ბეჭდავს მორიგ სიმბოლოს, ხოლო
მეორე აუქმებს რამდენიმე უკანასკნელ ბრძანებას. განსაკუთრებულ
ყურადღებას იქცევს "მწერალი კიბორჩხალას" ერთი თავისებურება
გაუქმების ბრძანების შესრულებისას: გაუქმების ბრძანება, როგორც
ნებისმიერი სხვა ბრძანება, შეიძლება თავადაც გაუქმდეს.

ამოცანა

თქვენ უნდა მოახდინოთ "მწერალი კიბორჩხალის" პროგრამული
რეალიზაცია: პროგრამამ უნდა დაიწყოს ცარიელი სტრიქონიდან, მიიღოს
მომხმარებლის მიერ განსაზღვრული ბრძანებების მიმდევრობა და პასუხი
გასცეს შეკითხვებს ზოგიერთი პოზიციის შესახებ ტექსტის მიმდინარე
მდგომარეობისათვის შემდეგი წესებით:

- `Init()` — გამოიძახება ერთხელ - პროგრამის მუშაობის დაწყებისას,
არგუმენტების გარეშე. შეიძლება გამოყენებულ იქნას მონაცემთა
სტრუქტურის ინიციალიზაციისათვის. ამ ბრძანების გაუქმება არ
შეიძლება.
- `TypeLetter(L)` — ამატებს ტექსტის ბოლოში ქვედა რეგისტრის ერთ `L`
სიმბოლოს დიაპაზონიდან `a, ..., z`.
- `UndoCommands(U)` — აუქმებს უკანასკნელ `U` ცალ ბრძანებას, სადაც `U`
დადებითი მთელი რიცხვია..
- `GetLetter(P)` — ტექსტის მიმდინარე მდგომარეობისთვის აბრუნებს `P`
პოზიციაში მდგომ სიმბოლოს არაუარყოფითი `P` ინდექსისათვის.
პირველ სიმბოლოს ტექსტში აქვს ნულვანი ინდექსი. ეს შეკითხვა არ
წარმოადგენს ბრძანებას, ამიტომ გაუქმების ბრძანება მას არ ეხება.

`Init()` პროცედურის გამოძახების შემდეგ სხვა პროცედურების
გამოძახებათა რაოდენობა შეიძლება იყოს 0 ან მეტი, ნებისმიერი
თანმიმდევრობით. გარანტირებულია, რომ `U` არ აღემატება მანამდე

მიღებული ბრძანებების რაოდენობას, ხოლო P ნაკლებია გექსტის მიმდინარე სიგრძეზე (ანუ მიმდინარე ტექსტში სიმბოლოების რაოდენობაზე).

ბრძანება UndoCommands(U) აუქმებს წინა U ცალ ბრძანებას შეზრუნებული თანმიმდევრობით: თუ გაუქმებული ბრძანებაა TypeLetter(L), მაშინ სიმბოლო L წაიშლება გექსტის ბოლოში; თუ გაუქმებული ბრძანებაა UndoCommands(X) X-ის რაღაც მნიშვნელობისათვის, მაშინ ისევ ხდება წინა X ცალი ბრძანების გამოყენება თავდაპირველი თანმიმდევრობით.

მაგალითი

ქვემოთ მოყვანილია გამოცხებათა თანმიმდევრობა გექსტის მდგომარეობებთან ერთად ყოველი გამოცხების შემდეგ.

გამოცხება	შედეგი	მიმდინარე ტექსტი
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

ქვეამოცანა 1 [5 ქულა]

- ბრძანებათა და შეკითხვათა საერთო რაოდენობა 1-დან 100-მდეა (ჩათვლით) და არ იქნება UndoCommands ბრძანების გამოცხება.

== ქვეამოცანა 2 [7 ქულა] ==

- ბრძანებათა და შეკითხვათა საერთო რაოდენობა 1-დან 100-მდეა (ჩათვლით) და არ იქნება UndoCommands ბრძანების გაუქმება.

ქვეამოცანა 3 [22 ქულა]

- ბრძანებათა და შეკითხვათა საერთო რაოდენობა 1-დან 5000-მდეა (ჩათვლით).

ქვეამოცანა 4 [26 ქულა]

- ბრძანებათა და შეკითხვათა საერთო რაოდენობა 1-დან 1 000 000-მდეა (ჩათვლით). GetLetter ბრძანების ყველა გამოძახება მოხდება TypeLetter და UndoCommands ბრძანებების გამოძახების შემდეგ.

ქვეამოცანა 5 [40 ქულა]

- ბრძანებათა და შეკითხვათა საერთო რაოდენობა 1-დან 1 000 000-მდეა (ჩათვლით).

რეალიზაციის დეტალები

თქვენ უნდა ჩააბაროთ მხოლოდ ერთი ფაილი, სახელად `scrivener.c`, `scrivener.cpp` ან `scrivener.pas`. ამ პროგრამამ უნდა მოახდინოს ზემოთ აღწერილი პროცედურების გამოძახება შემდეგი სიგნატურით.

C/C++ პროგრამებისათვის

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

Pascal პროგრამებისათვის

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

თქვენი ქვეპროგრამები (ფუნქციები) უნდა მოქმედებდნენ ისე, როგორც ეს ზემოთაა აღწერილი. ცხადია, თქვენ შეგიძლიათ გამოიყენოთ სხვა ფუნქციებიც შიგა მოხმარებისათვის, თქვენი ამოხსნა არ უნდა უნთიერებდეს სტანდარტულ შეჯანა-გამოჯანასთან, ასევე სხვა ფაილებთან.

შეფასების მაგალითი

წარმოდგენილი შემფასებელი მოდული კითხულობს შესაგან მონაცემებს შემდეგი ფორმატით:

- სტრიქონი 1: ბრძანებათა და შეკითხვათა საერთო რაოდენობა;
- ყოველი მომდევნო სტრიქონისათვის:
 - T, რომლის შემდეგ ჰყავს და ქვედა რეგისტრის სიმბოლო ბრძანებისათვის TypeLetter command;
 - U, რომლის შემდეგ ჰყავს და მთელი რიცხვი ბრძანებისათვის UndoCommands;
 - P, რომლის შემდეგ ჰყავს და მთელი რიცხვი ბრძანებისათვის GetLetter.

წარმოდგენილი შემფასებელი მოდული ბეჭდავს `GetLetter` ფუნქციის მიერ
დაბრუნებულ სიმბოლოებს - თითოეულ სტრიქონში.