

ماشین تحریر خرچنگ

بعضی از مردم معتقد هستند که «لئوناردو» یکی از طرفداران پر و پا قرص «یوهان گوتنبرگ» - آهنگر آلمانی که ماشین چاپ متحرک را اختراع کرد - بوده است، تا حدی که یک دستگاه که آن را «ماشین تحریر خرچنگ» می نامند طراحی کرد که در حقیقت یک ابزار تایپ بسیار ساده است. این دستگاه شبیه یک ماشین تایپ مدرن است و فقط دو دستور را می پذیرد: یکی برای تایپ حرف بعدی، و دیگری برای باطل (Undo) کردن دستورهای اخیر. خصوصیت قابل توجه ماشین تحریر خرچنگ آن است که دستور Undo خیلی قدرتمند است: یک Undo خودش هم به عنوان یک دستور تلقی میشود، و میتواند با دستور Undo دیگری ابطال گردد.

شرح مساله

شما میبایست یک نسخه نرم افزاری از ماشین تحریر خرچنگ را ایجاد کنید که با یک متن خالی شروع میکند، و به عنوان ورودی این موارد را دریافت میکند: یک توالی دستور که توسط کاربر وارد میشود، و همچنین جستار (Query) هایی برای تعیین حرف قرار گرفته در موقعیت های مشخصی از نسخه فعلی متن، به شرحی که گفته میشود.

- `Init()` — یک بار در شروع اجرای برنامه بدون هیچ پارامتری فراخوانی میشود. این دستور میتواند برای مقداره‌ی اولیه داده ساختارها مورد استفاده قرار گیرد. این دستور هرگز مورد ابطال (Undo) قرار نمیگیرد.

- `TypeLetter(L)` — یک حرف کوچک L را که از بین حرفهای a, ..., z انتخاب میشود به انتهای متن اضافه میکند.

- `UndoCommands(U)` — به ازای عدد صحیح مثبت U، آخرین U دستور را ابطال میکند.

- `GetLetter(P)` — به ازای اندیس نامنفی P، حرف قرار گرفته در موقعیت P از نسخه کنونی متن را برمیگرداند. اولین حرف از متن اندیس صفر دارد. این جستار یک دستور نیست، لذا دستور ابطال (Undo) آن را نادیده می گیرد.

بعد از فراخوانی اولیه دستور `Init()`، سایر تابعها میتوانند صفر مرتبه یا بیشتر به هر ترتیب دلخواهی فراخوانی شوند. تضمین میگردد که U هیچ وقت از تعداد دستورهایی که قبلاً دریافت شده اند بیشتر نخواهد بود، و P همواره کمتر از طول (تعداد حروف) متن کنونی خواهد بود.

دستور `UndoCommands(U)`، تعداد U دستور قبل را به ترتیب برعکس ابطال میکند: اگر دستوری که قرار است باطل شود `TypeLetter(L)` باشد، این دستور حرف L را از انتهای متن کنونی حذف میکند. اگر دستوری که قرار است باطل شود `UndoCommands(X)` به ازای مقداری از X باشد، تعداد X دستور قبل از آن به ترتیب اصلی‌شان بازیابی (Redo) میشوند.

مثال

ما یک توالی فراخوانی ها را به همراه وضعیت متن بعد از هر فراخوانی نمایش میدهیم.

متن کنونی	مقدار بازگشتی	فراخوانی
		Init()
a		TypeLetter(a)
ab		TypeLetter(b)
ab	b	GetLetter(1)
abd		TypeLetter(d)
a		UndoCommands(2)
abd		UndoCommands(1)
abd	d	GetLetter(2)
abde		TypeLetter(e)
abd		UndoCommands(1)
ab		UndoCommands(5)
abc		TypeLetter(c)
abc	c	GetLetter(2)
abd		UndoCommands(2)
abd	d	GetLetter(2)

زیر مساله ۱ [۵ نمره]

تعداد کل دستورها و جستارها از ۱ تا ۱۰۰ میباشد و دستور UndoCommands فراخوانی نمیگردد.

زیر مساله ۲ [۷ نمره]

تعداد کل دستورها و جستارها از ۱ تا ۱۰۰ میباشد و هیچ دستور UndoCommands ای ابطال (Undo) نمیگردد.

زیر مساله ۳ [۲۲ نمره]

تعداد کل دستورها و جستارها از ۱ تا ۵۰۰۰ خواهد بود.

زیر مساله ۴ [۲۶ نمره]

تعداد کل دستورها و جستارها از ۱ تا ۱۰۰۰۰۰۰۰ میباشد. همه فراخوانیهای تابع GetLetter بعد از همه فراخوانیهای توابع TypeLetter و UndoCommands خواهد بود.

زیر مساله ۵ [۴۰ نمره]

تعداد کل دستورها و جستارها از ۱ تا ۱۰۰۰۰۰۰۰۰ میباشد.

جزئیات پیاده سازی

شما باید دقیقا یک فایل را ارسال کنید که scrivener.cpp، scrivener.c یا scrivener.pas نام دارد. این فایل میبایست حاوی پیاده سازی زیر برنامه های فوق با قالب زیر باشد.

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

برنامه های پاسکال

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

این زیر برنامه ها باید طبق آنچه در بالا گفته شد رفتار کنند. البته شما میتوانید زیر برنامه های دیگری برای استفاده داخلی بنویسید. ارسال های شما نباید هیچ تعاملی با ورودی/خروجی استاندارد یا هیچ فایل دیگری داشته باشند.

ارزیابی نمونه

سیستم ارزیابی نمونه، ورودی را طبق قالب زیر میخواند:

- سطر ۱: تعداد کل دستورها و جستارهای ورودی
- در هر یک از سطرهای بعدی:
 - T و پس از آن یک فاصله و یک حرف کوچک برای دستور TypeLetter
 - U و پس از آن یک فاصله و یک عدد صحیح برای دستور UndoCommands
 - P و پس از آن یک فاصله و یک عدد صحیح برای دستور GetLetter

سیستم نمره دهی کاراکترهایی که توسط GetLetter برگردانده شوند را - هر کدام در یک خط - چاپ میکند