

بررسی پرانتزهای متوازن

صدری میخواهد یک ماشین حساب طراحی کند که باید تشخیص دهد آیا پرانتزهایی که کاربر در یک عبارت ریاضی وارد می‌کند متوازن هستند یا خیر. به این معنی که اگر پرانتزی باز می‌شود، باید یک پرانتز بسته هم برایش وجود داشته باشد و جای آن‌ها هم درست باشد. پرانتزها میتوانند از سه نوع $()$ ، $\{\}$ ، و $[]$ باشند. او از شما کمک می‌خواهد تا راهنمایی‌اش کنید که چگونه از پشته برای این مسأله استفاده کند.

مثال اول

ورودی

- $\{ \{ () \} \}$

خروجی

- True

مثال ۲

ورودی

- $\{ \{ () \} \}$

خروجی

- False

پاسخ مورد انتظار برای این مسئله شبه کد میباشد

معکوس کردن صف

علی به تازگی در یک شرکت برنامه‌نویسی مشغول به کار شده است. در یکی از روزهای کاری، مدیر او به سراغش آمد و مشکلی را با او در میان گذاشت. مدیر توضیح داد که در برخی از سیستم‌های صف‌بندی شرکت، نیاز است تا ترتیب چند مورد اول صف برعکس شود. برای مثال، اگر در صف مشتری‌ها چند نفر اول به دلایلی باید ابتدا رسیدگی شوند اما ترتیبشان معکوس شود، علی باید راه‌حلی برای این موضوع پیدا کند.

مدیر اضافه کرد که علی می‌تواند از استک و صف برای این کار استفاده کند. علی که به تازگی در کلاس ساختمان داده با استک‌ها و صف‌ها آشنا شده بود، سعی کرد راه‌حلی پیدا کند که بتواند با استفاده از این دو مفهوم اولین k نفر را در صف برعکس کند، بدون اینکه ترتیب بقیه اعضای صف به هم بخورد.

او از شما کمک می‌خواهد تا برای حل این مسئله شبه کدی به او پیشنهاد دهید که بتواند این تغییر را اعمال کند.

مثال اول

ورودی

- $Q = [1, 2, 3, 4, 5, 6, 7]$, $k = 4$

خروجی

- $Q = [4, 3, 2, 1, 5, 6, 7]$

مثال دوم

ورودی

- $Q = [10, 20, 30, 40, 50]$, $k = 3$

خروجی

- $Q = [30, 20, 10, 40, 50]$

پاسخ مورد انتظار برای این مسئله شبه کد میباشد

تبدیل مسیر

**** شما یک مسیر مطلق برای یک سیستم فایل سبک یونیکس دریافت می‌کنید که همیشه با یک اسلش / شروع می‌شود. وظیفه شما تبدیل این مسیر مطلق به مسیر استاندارد و ساده‌شده‌ی آن است.**

قوانین یک سیستم فایل سبک یونیکس به شرح زیر است:

- یک نقطه . نشان‌دهنده‌ی دایرکتوری فعلی است.
- دو نقطه .. نشان‌دهنده‌ی دایرکتوری والد (یک سطح بالاتر) است.
- چندین اسلش متوالی مانند // و /// به یک اسلش / تبدیل می‌شوند.
- هر توالی نقطه‌ای که مطابق قوانین بالا نباشد، باید به‌عنوان نام معتبر دایرکتوری یا فایل در نظر گرفته شود. به‌عنوان مثال، ... و نام‌های معتبر دایرکتوری یا فایل هستند.

مسیر استاندارد و ساده‌شده باید مطابق با این قوانین باشد:

- مسیر باید با یک اسلش / شروع شود.
- دایرکتوری‌ها در مسیر باید دقیقاً با یک اسلش / جدا شوند.
- مسیر نباید با اسلش / پایان یابد، مگر اینکه دایرکتوری ریشه باشد.
- مسیر نباید حاوی نقطه‌های تکی یا دوتایی (.) و (..) برای نشان‌دادن دایرکتوری فعلی یا والد باشد.

شما باید مسیر ساده‌شده‌ی استاندارد را برگردانید.

مثال‌ها

مثال ۱:

ورودی:

/home/

خروجی:

/home

توضیح: اسلش پایانی باید حذف شود.

مثال ۲:

ورودی:

/home//foo/

خروجی:

/home/foo

توضیح: چندین اسلش متوالی به یک اسلش تبدیل می‌شوند.

مثال ۳:

ورودی:

/home/user/Documents/../Pictures

خروجی:

/home/user/Pictures

توضیح: دو نقطه .. به سطح بالاتر از دایرکتوری اشاره دارد.

مثال ۴:

ورودی:

/. ./

خروجی:

/

توضیح: رفتن به یک سطح بالاتر از ریشه امکان پذیر نیست.

مثال ۵:

ورودی:

/.../a/./b/c/./d/./

خروجی:

/.../b/d

توضیح: ... به عنوان یک نام معتبر برای دایرکتوری در نظر گرفته می شود و سایر .. و . ها به دایرکتوری معادل آن تبدیل می شوند.

محدودیت ها

- طول رشته ورودی path بین 1 تا 3000 است.
- path شامل حروف انگلیسی، اعداد، نقطه ،، اسلش / یا زیر خط _ است.
- path یک مسیر مطلق معتبر در یونیکس است.

**

رونمایی از کارت‌ها به ترتیب صعودی

****** به شما یک آرایه از اعداد صحیح به نام deck داده شده است. deck یک دسته از کارت‌ها است که روی هر کارت یک عدد منحصر به فرد وجود دارد. عدد روی کارت نام برابر با `deck[i]` است.

شما می‌توانید این دسته کارت‌ها را به هر ترتیبی که می‌خواهید مرتب کنید. در ابتدا، تمام کارت‌ها به صورت رو به پایین (پنهان) در یک دسته قرار دارند.

شما مراحل زیر را به طور مکرر انجام می‌دهید تا تمام کارت‌ها رونمایی شوند:

۱. کارت بالایی دسته را بردارید، آن را نمایش دهید و از دسته خارج کنید.
 ۲. اگر هنوز کارت‌هایی در دسته وجود دارند، کارت بالایی بعدی را به پایین دسته منتقل کنید.
 ۳. اگر هنوز کارت‌های پنهان وجود دارند، به مرحله ۱ برگردید. در غیر این صورت، متوقف شوید.
- یک ترتیبی از دسته کارت‌ها را برگردانید که کارت‌ها را به ترتیب صعودی نمایش دهد.

توجه داشته باشید که اولین عنصر در جواب، به عنوان بالای دسته در نظر گرفته می‌شود.

مثال‌ها

مثال 1:

ورودی:

7
17 13 11 2 3 5 7

خروجی:

2 13 3 11 5 17 7

توضیح:دسته کارت‌ها را به ترتیب [17,13,11,2,3,5,7] دریافت می‌کنیم (این ترتیب مهم نیست) و آن را بازآرایی می‌کنیم.پس از بازآرایی، دسته به شکل [2,13,3,11,5,17,7] شروع می‌شود، جایی که 2 بالای دسته است.ما 2 را رونمایی می‌کنیم و 13 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [3,11,5,17,7,13] است.ما 3 را رونمایی می‌کنیم و 11 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [5,17,7,13,11] است.ما 5 را رونمایی می‌کنیم و 17 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [7,13,11,17] است.ما 7 را رونمایی می‌کنیم و 13 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [11,17,13] است.ما 11 را رونمایی می‌کنیم و 17 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [13,17] است.ما 13 را رونمایی می‌کنیم و 17 را به پایین منتقل می‌کنیم. دسته اکنون به شکل [17] است.ما 17 را رونمایی می‌کنیم.از آنجایی که تمام کارت‌های رونمایی شده به ترتیب صعودی هستند، جواب درست است.

مثال 2:

ورودی:

2
1 1000

خروجی:

1 1000

محدودیت‌ها

- طول آرایه حداکثر 1000 است.
- مقدار $deck[i]$ بین 1 تا 106 است. همه مقادیر deck منحصر به فرد هستند.

diGital world | امتیازی

یک روز در دنیای دودویی، یک جمعیت فوق العاده عجیبی از اعداد دیجیتال زندگی میکردند. این اعداد دیجیتال تنها دارای دو رقم، 0 و 1 بودند و همیشه به دنبال ماجراهای دیجیتالی بودند.

در این دنیای عجیب، یک دیجیتال به نام دیجیتال جوان به دنیا آمد. دیجیتال جوان به تازگی توانسته بود یک مهارت جدید را بیاموزد:

تولید اعداد دودویی از 1 تا N

به عنوان مثال، اگر N برابر با 3 باشد، دیجیتال جوان ابتدا عدد 1 تولید میکند، سپس عدد 10 و در نهایت عدد 11. او این کار را با مهارت و توانایی بالایی انجام میدهد و همیشه اعداد دودویی را با بالاترین دقت تولید میکند.

دیگیتالی جوان خوشحال است که میتواند به دیجیتالهای دیگر در دنیای دودویی آموزش دهد که چگونه اعداد دودویی از 1 تا N را تولید کنند. او به عنوان "دیگیتالی جوان و با مهارت تولید اعداد دودویی" به یاد میماند.

ورودی

در خط اول به شما عدد طبیعی n داده میشود.

$$1 \leq n \leq 1000$$

خروجی

در خروجی n عدد باینری اول را (بدون نمایش صفرهای اول عدد) نمایش دهید.

مثال

ورودی نمونه 1

2

خروجی نمونه 1

1

10

ورودی نمونه 2

5

خروجی نمونه 2

1

10

11

100

101