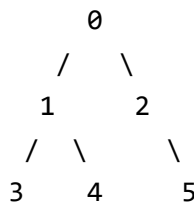


درخت بکاریم!

در این تمرین از شما می‌خواهیم تا ساختمان داده یک درخت باینری را طراحی کنید و در تمرین‌های بعدی مورد استفاده قرار دهید برای طراحی درختان به چند نکته توجه کنید:

- درخت شما می‌بایست به صورت تمپلیت باشد که تا بتوانید درختی از هر نوع کلاس را داشته باشید (درخت فقط توانایی هندل کردن char و یا هر یک از int , float , double را داشته باشد کافی است و نیازی به سایر کلاس‌ها نیست)
- شما نیاز به یک تابع تفسیر پیمایش بر روی درخت دارید که با دریافت ورودی به این صورت رشته ای از r , l رفرنسی از نود مورد نظر را برگرداند. مثلاً در این مثال با دریافت rr



باید رفرنسی به نود با شماره 5 برگردانده شود.

- تابع Add که مجدداً با دریافت یک ورودی از جنس اعضای درخت و یک رشته متشکل از r (نشان دهنده شاخه سمت راست) و l (نشان دهنده یک شاخه از سمت چپ) آن را در محل مناسب قرار می‌دهد
- تابع سرچ که با دریافت یک ورودی از جنس اعضای درخت بر روی درخت سرچ انجام داده و رفرنسی از عضو پیدا شده را برگرداند.
- تابع Remove که دو حالت دارد در صورتی که یک رشته از r , l دریافت کرد بر روی درخت پیمایش کرده تا به نود مورد نظر برسد و آن را حذف کند (دقت کنید هندل کرد وضعیت فرزندان نیز مهم از و آنان نباید حذف شوند)

+حالت دوم تابع ریمو یک ورودی از جنس اعضای درخت دریافت کرده و در درخت به دنبال آن عضو می‌گردد و آن را حذف می‌کند. (دقت کنید هندل کرد وضعیت فرزندان نیز مهم از و آنان نباید حذف شوند)

راه کج به منزل می‌رسد!



در این سوال طراح سوال می‌خواهد ثابت کند که راه کج به منزل می‌رسد! بنابراین از شما می‌خواهد تا به درختی که در سوال قبلی نوشته اید تابع $rahKaj$ رو اضافه کنید که این تابع به این صورت عمل می‌کند که درخت ما را به صورت مورب می‌پیماید و نودهای درخت را چاپ می‌کند.

ورودی

برای حل این سوال ابتدا نیاز است یک تابع درخت ساز به کدتان اضافه کنید که با توجه به ورودی‌های ذکر شده در ادامه درخت مورد نظر را ایجاد کند.

در خط اول عدد n که تعداد گره‌های درخت است به شما داده می‌شود.

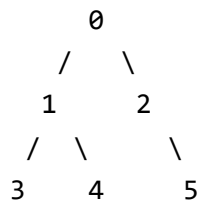
در $n - 1$ خط بعدی جایگاه هر گره (به جز ریشه) نسبت به ریشه به صورت رشته‌ای شامل دو کاراکتر 'r' و 'l' به شما داده می‌شود.

به عنوان مثال رشته "rl" به این معناست که گره ای در موقعیت فرزند چپ فرزند راست ریشه وجود دارد.

موقعیت گره ها به ترتیب سطحی به شما داده می شود.

مثلاً درخت متناظر با ورودی زیر به این صورت است:

6
1
r
11
1r
rr



برای درک بهتر پیمایش مورب به شکل زیر توجه کنید.(هر سطر به سورت مرتب شده چاپ میشود).

 توضیح تصویر

8 10 14
3 6 7 13
1 4

خروجی

پیمایش مورب درخت را چاپ کنید.

نمونه ورودی 1

6
1

r
ll
lr
rr

نمونه خروجی 1

0 2 5
1 4
3

راه کج به منزل نرسید: /



خب در سوال قبلی طی اتفاقی عجیب طراح متقاعد شد راه کج به منزل نرسید بنابراین در این سوال از شما می‌خواهد که تابع `rahRast` رو به درخت باینری تون اضافه کنید که در طی اون نودهای درخت به صورت عمودی چاپ شوند:

ورودی

مثل سوال قبل از همون تابع درخت ساز طبق ورودی های زیر استفاده کنید:

در خط اول عدد n که تعداد گره های درخت است به شما داده می شود.

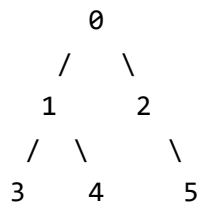
در $n - 1$ خط بعدی جایگاه هر گره (به جز ریشه) نسبت به ریشه به صورت رشته ای شامل دو کاراکتر 'r' و 'l' به شما داده می شود.

به عنوان مثال رشته "rl" به این معناست که گره ای در موقعیت فرزند چپ فرزند راست ریشه وجود دارد.

موقعیت گره ها به ترتیب سطحی به شما داده می شود.

مثلاً درخت متناظر با ورودی زیر به این صورت است:

6
l
r
ll
lr
rr



برای درک بهتر پیمایش عمودی به شکل زیر توجه کنید. (هر سطر به سورت مرتب شده چاپ میشود.)

 توضیح تصویر

خروجی

پیمایش عمودی درخت را چاپ کنید.

نمونه ورودی 1

6
1
r
11
1r
rr

نمونه خروجی 1

3
1
0 4
2
5

یک سوال سنگین

وزن سنگین ترین مسیر یک درخت باینری را بیابید.

ورودی

در خط اول عدد n که تعداد گره های درخت است به شما داده می شود.

در خط دوم وزن گره ریشه به شما داده می شود.

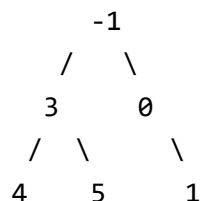
در $n - 1$ خط بعدی جایگاه هر گره (به جز ریشه) نسبت به ریشه به صورت رشته ای شامل دو کاراکتر 'r' و 'l' به شما داده می شود. همچنین وزن هر گره نیز به شما داده می شود.

به عنوان مثال رشته "rl" به این معناست که گره ای در موقعیت فرزند چپ فرزند راست ریشه وجود دارد.

موقعیت گره ها به ترتیب سطحی به شما داده می شود.

مثلاً درخت متناظر با ورودی زیر به این صورت است:

```
6
-1
l 3
r 0
ll 4
lr 5
rr 1
```



خروجی

وزن سنگین ترین مسیر را چاپ کنید.

نمونه ورودی 1

```
6
-1
1 3
r 0
ll 4
lr 5
rr 1
```

نمونه خروجی 1

```
12
```

یک سوال سبک (اختیاری)

پیمایش Preorder یک درخت باینری را از پیمایش های Inorder و Postorder آن به دست آورید.

ورودی

در خط اول تعداد گره های درخت به شما داده می شود.

در خط دوم پیمایش Inorder و در خط سوم پیمایش Postorder به شما داده می شود.

خروجی

پیمایش Preorder درخت را چاپ کنید.

نمونه ورودی 1

```
6
3 1 4 0 2 5
3 4 1 5 2 0
```

نمونه خروجی 1

```
0 1 3 4 2 5
```