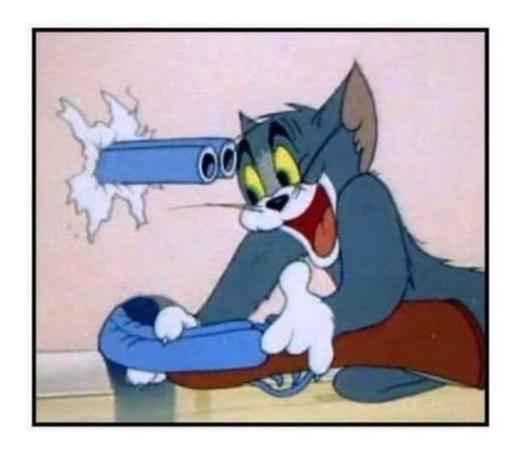
بچرخ تا بچرخیم

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



لورا بعد از کار با ساختمان داده صف، حسابی از آن خوشش آمده. اما او متوجه ضعفی در ساز و کار این ساختمان داده شده است. و آن این است که اگر ما k عنصر را وارد صف کنیم و سپس آنها را یکی یکی حذف کنیم شرط پر بودن صف برقرار میماند و این در حالی است که صف هنوز جای خالی دارد ولی ما نمیتوانیم از آن فضاهای خالی استفاده کنیم!

او در تلاش است ساختمان داده صف را به گونهای پیادهسازی کند که این ضعف را برطرف کند و بتواند از فضاهای خالی جلوی صف برای ذخیره مقادیر جدید استفاده کند.

لورا در نظر دارد پیادهسازی را به گونهای انجام دهد که از عملیات زیر پشتیبانی کند:

▼ MyCustomizeQueue(k)

سازندهای است که اندازه صف را k ست میکند.

▼ front()

تابعی است که داده جلوی صف را برمیگرداند. درصورت خالی بودن صف، 1- برمیگرداند.

▼ rear()

تابعی است که آخرین داده صف را برمیگرداند . درصورت خالی بودن صف 1- برمیگرداند.

▼ enqueue(value)

تابعی است که داده value را در صف قرار میدهد. درصورتی که عملیات با موفقیت انجام شود true برمیگرداند.

▼ dequeue()

تابعی است که براساس FIFO داده از صف حذف میکند. درصورتی که عملیات با موفقیت انجام شود true

▼ isEmpty()

تابعی است که بررسی میکند صف خالی است یا نه. درصورت خالی بودن و در غیر اینصورت false و در غیر اینصورت false

▼ isFull()

تابعی است که بررسی میکند صف پر است یا خیر. درصورت خالی بودن false و در غیر اینصورت true

از آنجایی که شما هم با صف آشنایی دارید؛ به لورا در پیادهسازی این ساختمان داده کمک کنید.

ورودي

خط اول ورودی شامل k، سایز صف، خط دوم ورودی شامل n، و در n خط بعدی یکی از چند توابع بالا وارد می شود.

خروجی

خروجی شامل n خط است که خروجی تابع متناظر با هر عملیات در هر خط چاپ می شود.

ورودی نمونه ۱

3
10
enQueue 1
enQueue 2
enQueue 3
enQueue 4
Rear
isFull
deQueue
enQueue 5
Rear
isEmpty

خروجی نمونه ۱

true
true
false
3
true
true
true
5
false