

آرایه کاغذی

به سوالات کاغذی آرایه ما جواب دهید تا دستتون گرم بشه

1.1) نوع داده ای مجرد یا انتزاعی چیست (Abstract Data Type) ؟

1.2) یک نوع داده‌ای انتزاعی (ADT) به نام "Fraction" طراحی کنید که قابلیت نمایش و انجام عملیات ریاضی روی اعداد کسری را داشته باشد. این ADT باید شامل ویژگی‌های زیر باشد:

۱. ویژگی‌های مورد نیاز (Attributes):

- صورت (Numerator)
- مخرج (Denominator)

۲. عملیات مورد نیاز (Operations):

- ☒ ایجاد یک کسر (ساخت یک نمونه از Fraction)
- ☒ نمایش کسر به صورت رشته (مثل "3/4")
- ☒ جمع دو کسر (Fraction + Fraction)
- ☒ تفریق دو کسر (Fraction - Fraction)
- ☒ ضرب دو کسر (Fraction * Fraction)
- ☒ تقسیم دو کسر (Fraction ÷ Fraction)
- ☒ ساده‌سازی خودکار کسر (مثلاً تبدیل "6/8" به "3/4")
- ☒ تبدیل کسر به عدد اعشاری

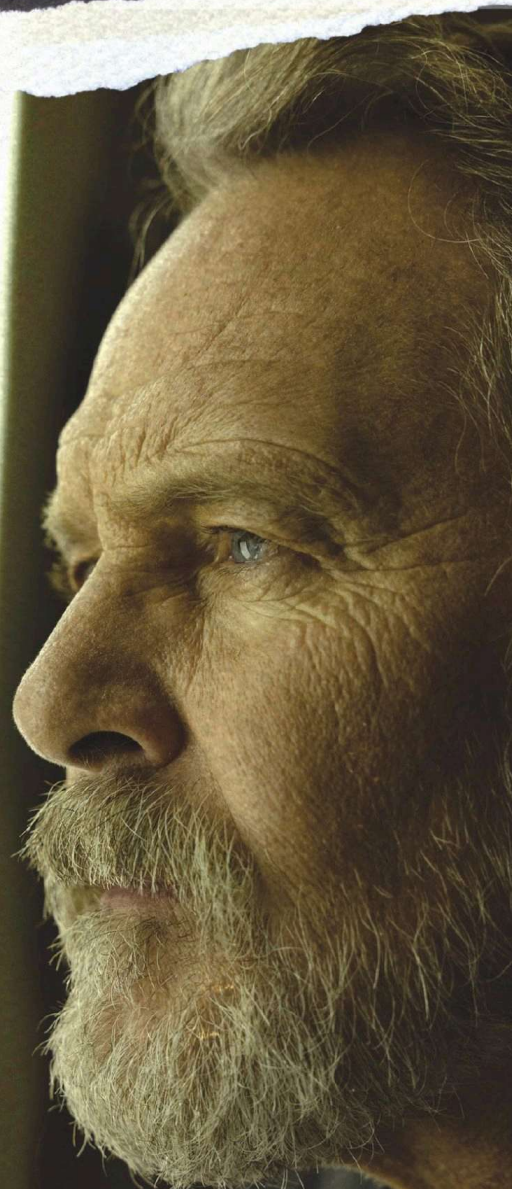
از آرایه یا struct (در ++C) استفاده کنید. **برنامه‌ای بنویسید که دو عدد کسری از کاربر بگیرد و عملیات بالا را انجام دهد. (روی کاغذ)

2) (ترکیب جستجو و مرتب‌سازی) فرض کنید یک آرایه نامرتب بسیار بزرگ دارید و می‌خواهید در آن جستجو کنید. آیا بهتر است ابتدا آن را مرتب کنید و سپس جستجوی باینری انجام دهید، یا مستقیماً جستجوی خطی انجام دهید؟ چه عواملی در این انتخاب تأثیر دارند؟

3) یک آرایه سه بعدی $100 \times 100 \times 100$ به صورت **سطری** ذخیره شده است. آدرس عنصر $arr[0][0]$ $[0]$ برابر است با H (یک عدد نامشخص) اگر عناصری که در این آرایه ذخیره می‌شود از نوع *Integer* باشد. آدرس عنصر $arr[25][60][48]$ را بدست آورید. (با فرض اینکه هر عدد صحیح (Integer) برابر 2 بایت فضا اشغال کند.)

فراتر از یک کاغذ

I am
more
than
who
I am
on
paper.



تو بهش بگو آرایه من میگم عشق

به سوالات زیر به صورت خلاصه و کوتاه پاسخ دهید. توجه داشته باشید که ارائه ی دلیل برای هر سوال اجباریست!

1) یک آرایه به طول n داریم. چگونه می‌توان بدون استفاده از حلقه، مقدار تمام خانه‌های آن را مقداردهی اولیه کرد؟ آیا این روش برای آرایه‌های بسیار بزرگ بهینه است؟ چرا؟

2) اگر آرایه‌ای مرتب‌شده داشته باشیم و بخواهیم یک عنصر جدید را در آن درج کنیم، پیچیدگی زمانی این عملیات در بدترین حالت چقدر است؟ آیا راهی برای بهینه‌سازی این فرآیند وجود دارد؟

3) اگر آرایه یک ساختار داده است که مقادیر را در مکان‌های متوالی حافظه ذخیره می‌کند، پس چرا گاهی دسترسی به یک عنصر کندتر از $O(1)$ می‌شود؟

4) چرا آرایه‌های دو بعدی در بعضی زبان‌ها (مثلاً C) به صورت "row-major" ذخیره می‌شوند؟ اگر برعکس بود (یعنی "column-major") چه تأثیری بر عملکرد برنامه داشت؟

5) چرا در بعضی موارد تخصیص حافظه آرایه ممکن است خطای "Out of Memory" بدهد، حتی اگر حافظه کافی به نظر برسد؟

پیچیدگی آرایه ای

یک **آرایه پویا** (Dynamic Array) را در نظر بگیرید که ظرفیت آن در صورت پر شدن، دو برابر می‌شود. شما باید پیچیدگی زمانی عملیات‌های زیر را در **بهترین، بدترین و میانگین حالت** تحلیل کنید:

- 1 درج یک عنصر در انتهای آرایه
- 2 درج یک عنصر در ابتدای آرایه
- 3 درج یک عنصر در موقعیت دلخواه (index i)
- 4 حذف یک عنصر از انتهای آرایه
- 5 حذف یک عنصر از ابتدای آرایه
- 6 حذف یک عنصر از موقعیت دلخواه (index i)
- 7 دسترسی به یک عنصر با اندیس i
- 8 جستجوی یک مقدار در آرایه (حالت تریبی و دودویی)
- 9 مرتب‌سازی آرایه (حالت بهترین و بدترین برای الگوریتم‌های مختلف بگویید)

نکات: 💡

- در پاسخ خود، پیچیدگی زمانی هر عملیات را به صورت **Big-O** تحلیل کنید.
- برای درج و حذف در آرایه پویا، نحوه تخصیص مجدد حافظه را نیز در نظر بگیرید.

سوال:

- اگر از آرایه ثابت (Static Array) به جای آرایه پویا استفاده شود، کدام عملیات‌ها دچار تغییر در پیچیدگی می‌شوند؟ چرا؟

بازی مرکب

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

شما سربازی در بازی مرکب هستید؛ در این مرحله به ترتیب بازیکنان از داخل یک جعبه، گوی هایی را برمیدارند **(بر روی هر گوی عددی وجود دارد)**؛ قانون این بازی به اینصورت است که اگر مجموع اعداد گوی های دو فرد برابر با عدد بر روی **تابلو** شود کل گروه محکوم به **مرگ** می شوند. به **پیشرو (Front Man)** کمک کنید تا به هدفش برسد.

▼ بیا راحت کنم :

یک آرایه به نام `arr[]` شامل n عدد صحیح و یک مقدار هدف داده شده است. وظیفه این است که بررسی کنید آیا در این آرایه جفتی از عناصر وجود دارد که مجموع آن‌ها برابر با مقدار هدف باشد یا خیر.

ورودی

ورودی شامل دو خط است که در آن دو عدد طبیعی n و $Target$ با فاصله از هم آمده است و در خط دوم آرایه ای از اعداد به طول n دریافت می شود.

$$1 \leq n, Target \leq 100$$

خروجی

در تنها خط خروجی اگر دو عنصری پیدا شد که مجموعش با عدد $Target$ برابر شود `true` را چاپ کن در غیراینصورت `false`.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

5 -2
0 -1 2 -3 1

خروجی نمونه ۱

true

ورودی نمونه ۲

5 0
1 -2 1 0 5

خروجی نمونه ۲

false

زندگی



آرایه باینری

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

رشته ای باینری داده شده است. بیشترین تعداد **یک های متوالی** در این رشته را بازگردانید.

ورودی

در تنها خط ورودی **رشته ای باینری** (شامل 0 و 1) آمده است.

خروجی

در تنها خط خروجی بیشترین تعداد یک های متوالی را چاپ کنید.

ورودی نمونه ۱

110111

خروجی نمونه ۱

3

ورودی نمونه ۲

101101

خروجی نمونه ۲

همونی که خیلی می گنش



All for one and one for all,
united we stand divided we fall.

~ Alexandre Dumas

حلزون سرگردان

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

خانه **حلزونی** در مرکز یک **هزارتوی مارپیچی** (تنها مسیری که برای رسیدن به خانه اش دارد یک مسیر مارپیچ است) قرار دارد و **مسیر خانه اش (از دروازه ورودی هزارتو تا مرکز هزارتو)** با اعدادی مشخص شده اند. جادوگر جنگل به او گفت اگر بتواند همه ی اعداد مسیر خانه اش را برای جادوگر یادداشت کند می تواند یک دروازه تلپورت برای آن بسازد. به حلزون کمک کنید تا اعداد در مسیرش را یادداشت کند.

ورودی

ورودی خط اول دو عدد طبیعی row و $column$ است که سطر و ستون ماتریس است. و در خط های بعدی باید عناصر ماتریس را وارد کنید. در هر خط باید یک سطر از ماتریس وارد شود.

$$1 \leq row, column \leq 100$$

خروجی

خروجی برنامه ی شما باید شامل یک خط حاوی پیمایش مارپیچ عناصر ماتریس باشد.

خروجی نمونه ۱

```
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

##خروجی نمونه ۱

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

در خط اول 4 4 سطر و ستون ماتریکس هستند و در خطوط بعدی عناصر هر سطر ماتریس وارد شده

ورودی نمونه ۲

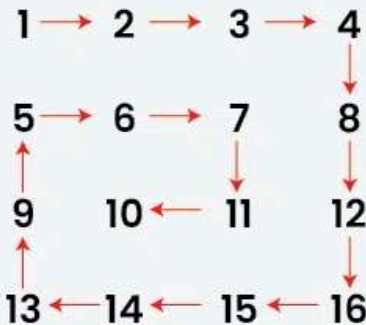
3 6
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18

خروجی نمونه ۲

1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11

نمونه ۱

Example of matrix in spiral form



Matrix:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Output: 1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10

حلزون سرگردان مونوگاتاری 物語



حداکثر حجم سیمان

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

مهندس عمرانی تصمیم می گیرد برای افزایش مقاومت یک سازه فضای خالی بین ستون های آن را با سیمان پر کند؛ به او کمک کنید تا بیشترین حجم سیمان مورد نیازش را پیدا کند.

فرض کنید سازه یک بعدی هست و اعدادی که برای نشان دادن طول ستون ها و حجم سیمان داریم؛ اعداد طبیعی هستند.

ورودی

در خط اول عدد n گرفته میشود در خط دوم لیستی از اعداد به طول n که نشان دهنده طول ستون ها است.

خروجی

خروجی برنامه ی شما حجم سیمان است.

ورودی نمونه ۱

11

1 0 2 1 0 1 3 2 1 2 1

خروجی نمونه ۱

6

ورودی نمونه ۲

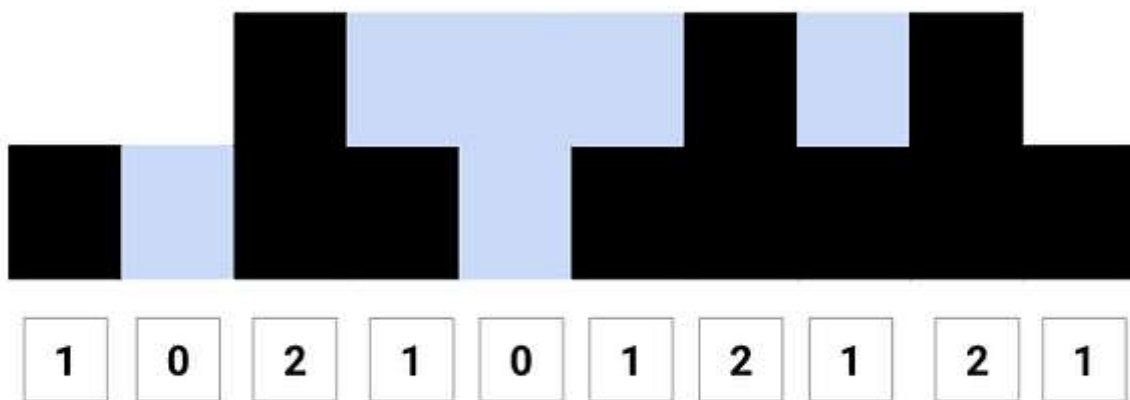
6

4 2 0 3 2 5

خروجی نمونه ۲

9

تصویر ورودی نمونه ۱



برنامه نویس کارگر



بزرگراه شهر عجایب

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

در شهر عجایب **آرمان** پاسبان بزرگراه **ماشین های عجیب** است در این بزرگراه سرعت هر خودرو باتوجه به قانون بزرگراه و دستور فرمانده **آلیس** به اینصورت مورد بررسی قرار می گیرد :

فرمانده آلیس دو عدد را به عنوان دستور به آرمان می دهد؛ عدد اول (n) تعداد خودرو هایی است که آرمان باید سرعت آن ها را مورد بررسی قرار دهد و اما عدد دوم (k)، قانون این بزرگراه به اینصورت است که پاسبان بزرگراه باید باتوجه به k از آخرین خودرویی که وارد بزرگراه شده به عنوان خودرو اول (تعداد ماشین های آرمان با ورودی اول محدود شده است) شروع کند و تا k خودرو بعدی را به عنوان یک دسته، سرعتشان را با هم مقایسه کند و در دفترش بیشترین سرعت را به عنوان سرعت غیر مجاز یادداشت کند و به همین صورت از ماشین دوم، سوم و تا جایی که k خودرو بعد از آن وجود داشته باشد شروع می کند به دسته بندی خودرو ها (**زیر آرایه ها با طول k**) و یادداشت سرعت غیر مجاز آن دسته در نهایت پاسبان آرمان یادداشت هایش را به آلیس **نمایش** می دهد.

توجه داشته باشید که باید از سمت چپ به راست آرایه مورد بررسی قرار گیرد.

▼ **بیا راحت کنم :**

یک آرایه به نام `arr[]` شامل n عدد صحیح و یک عدد صحیح k که نشان دهنده طول زیر آرایه ها است داده شده است. باید بیشینه ی عناصر زیر آرایه ها به طول k را پیدا کنی و در خروجی چاپ کنی.

ورودی

در خط اول دو عدد n و k گرفته می شود و در خط دوم لیستی از اعداد به طول n که آرایه مورد نظر ما است. (k) طول زیر آرایه های ما هست.)

خروجی

خروجی برنامه ی شما لیستی از بزرگترین عنصر هر زیر آرایه به طول k است.

ورودی نمونه ۱

6 3
1 2 3 1 4 5

خروجی نمونه ۱

3 3 4 5

Explanation : Maximum of 1, 2, 3 is 3 / Maximum of 2, 3, 1 is 3 / Maximum of 3, 1, 4 is 4
/Maximum of 1, 4, 5 is 5

ورودی نمونه ۲

10 4
8 5 10 7 9 4 15 12 90 13

خروجی نمونه ۲

10 10 10 15 15 90 90

Explanation : Maximum of 8, 5, 10, 7 is 10 / Maximum of 5, 10, 7, 9 is 10 / Maximum of 10, 7, 9, 4
is 10 / Maximum of 7, 9, 4, 15 is 15 / Maximum of 9, 4, 15, 12 is 15 / Maximum of 4, 15, 12, 90 is 90
/Maximum of 15, 12, 90, 13 is 90

بی مزه



شاخص ترازمندی

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

در یک آرایه **شاخص ترازمندی** شاخصی است که **جمع تمامی عناصر قبل از آن با جمع تمامی عناصر بعد از آن برابر باشد**. در صورتی که این اندیس وجود ندارد عدد -1 را چاپ کنید. (شاخص = اندیس = ایندکس)

ورودی

در خط اول عدد طبیعی t آمده است که نشان دهنده تعداد تست کیس ها است. در خط دوم عدد طبیعی n آمده است. در خط بعدی آرایه ای از اعداد به طول n آمده است. توجه داشته باشید پس از وارد کردن t به می توانید تست کیس ها را وارد کنید. این که خروجی هر تست کیس بعد از ورود آن چاپ شود مشکلی ایجاد نمی کند.

$$1 \leq n \leq 100$$

خروجی

به تعداد t خروجی داریم

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست ها داده می شود.

ورودی نمونه ۱

1 2 0 3

خروجی نمونه ۱

2

ورودی نمونه ۲

2
4
1 1 1 1
7
-7 1 5 2 -4 3 0

خروجی نمونه ۲

-1
3

تراز



گردان نخبگان(امتیازی)

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

فرمانده گردان نخبگان، **متین** وظیفه **مهمی** دارد؛

او هر روز صبح سربازانش را خبردار می کند و سپس دستور جا به جایی تعدادی از سربازان را بدون بهم ریختن ترتیبشان می دهد؛ سربازان وظیفه دارند صرفاً با شنیدن عددی که فرمانده (k) می گوید؛ بدون بهم ریختن ترتیبشان از جلوی صف خارج شده و به انتهای صف بپیوندند به صورتی که نفر اول صف به انتهای صف می پیوندد و به همین ترتیب به تعداد عددی که فرمانده گفته سربازان این کار را انجام می دهند. فرمانده متین وظیفه دارد باتوجه به صف جدید نام سربازان را از انتهای صف تا ابتدای آن در دفتر خاطرات مشکی اش (Console) نمایش دهد.

انتهای صف ایندکس صفر آرایه است. / نام سربازان در گردان با عدد مشخص شده است.

ورودی

در خط اول عدد n و k گرفته می شود؛ در خط دوم لیستی از اعداد به طول n (k دستور فرمانده است).

خروجی

خروجی برنامه ی شما باید شامل لیستی از اعداد (نام سربازان) باشد که فرمانده قصد دارد در دفتر بنویسد .

ورودی نمونه ۱

7 3
1 2 3 4 5 6 7

خروجی نمونه ۱

5 6 7 1 2 3 4

اعداد 5 6 7 بدون تغییر به انتها اضافه شده اند.

ورودی نمونه ۲

5 2
25 7 8 61 11

خروجی نمونه ۲

61 11 25 7 8

اعداد 61 11 بدون تغییر به انتها اضافه شده اند.

بیچارگان



"BY THE WAY, WHAT ARE WE IN LINE FOR?"