



## مقدمه

این تمرین برای آشنایی با برنامه‌نویسی بازگشتی طراحی شده است. این تمرین در قالب سه سوال مجزا تهیه شده است که پیشنهاد می‌شود برای درک بهتر مفاهیم برنامه‌نویسی بازگشتی، زمان کافی را برای پاسخ دادن به آن‌ها اختصاص دهید. توجه کنید که پرسش‌ها حتماً باید به روش بازگشتی حل شوند، هر چند ممکن است روش‌های دیگری نیز برای حل آن‌ها وجود داشته باشد. در سوال یک و دو استفاده از حلقه مجاز نیست و در سوال ۳ قسمت اصلی سوال (پیدا کردن مسیر پرش) حتماً باید به صورت بازگشتی پیاده‌سازی شود.

## پرسش‌ها

### ۱. خیابان فراموش شده

#### شرح مسئله

آکبرت که ساکن شهر نردها (Nerds City) است، در حال بازگشت به خانه خود است. ساختمان‌های هر خیابان از شهر نردها ویژگی‌های زیر را دارد:

۱. همه خیابان‌های شهر یک‌طرفه هستند.
۲. در هر خیابان دقیقاً یک ساختمان وجود دارد که از تمام ساختمان‌های دیگر در آن خیابان بلندتر است.
۳. دنباله ارتفاع ساختمان‌های قبل از ساختمان با ارتفاع پیشینه، اکیدا صعودی است.
۴. دنباله ارتفاع ساختمان‌های بعد از ساختمان با ارتفاع پیشینه، اکیدا نزولی است.

به بیان دیگر در هر خیابان دنباله‌ای به شکل زیر از ساختمان‌ها داریم که  $h[1]$  ارتفاع ساختمان اول در خیابان و  $h[n]$  ارتفاع آخرین ساختمان را نشان می‌دهد. در این دنباله ساختمان  $i$ م که بلندترین ساختمان خیابان است، ساختمان اوج نامیده می‌شود.

$$h[1] < h[2] < \dots < h[i-1] < h[i] > h[i+1] > \dots > h[n-1] > h[n]$$
$$1 \leq i \leq n$$

در شهر نردها وارد شدن به هر خیابان یک رمز می‌خواهد. رمز هر خیابان اندیس ساختمان با بیشینه ارتفاع (ساختمان اوج) است. آلبرت رمز خیابان خودشان را فراموش کرده و نمی‌تواند به خانه برگردد. پس از شما به‌عنوان یک برنامه‌نویس می‌خواهد تا به او کمک کنید رمز خیابان را بفهمد.

توجه کنید که ساختمان اوج می‌تواند ساختمان اول یا آخر خیابان نیز باشد.

## ورودی

- در سطر اول یک عدد ورودی  $n$  داده می‌شود که تعداد ساختمان‌های خیابان را مشخص می‌کند.
- در سطر دوم به تعداد  $n$  عدد صحیح داده می‌شود که  $i$ مین عدد  $h_i$  (ارتفاع ساختمان  $i$ ام) را نشان می‌دهد.
- تضمین می‌شود ورودی ویژگی گفته شده در صورت مسئله را دارا است.

## خروجی

در تنها خط خروجی، اندیس ساختمان با ارتفاع اوج را نمایش دهید (ساختمان‌ها از ۱ تا  $n$  شماره‌گذاری می‌شوند).

## محدودیت‌ها

- $1 \leq n \leq 10^6$
- $1 \leq h[i] \leq 10^9$

## ورودی و خروجی نمونه

ورودی	خروجی
5 1 4 5 7 2	4
6 5 20 19 17 13 1	2
2 1 2	2

## شرح ورودی و خروجی نمونه

در مثال اول، ساختمان با ارتفاع ۷ ساختمان اوج است که ساختمان شماره ۴ است.

در مثال دوم، ساختمان با ارتفاع ۲۰ ساختمان اوج است که ساختمان شماره ۲ است.

در مثال سوم، ساختمان با ارتفاع ۲ ساختمان اوج است که ساختمان شماره ۲ است.

## ۲. آلبرت فراموشکار

### شرح مسئله

هنگامی که آلبرت به خانه خود رسید، یادش افتاد که کلید خانه را فراموش کرده است. اما راهی دیگر برای ورود به خانه را به یاد داشت. او می‌تواند یک بازی با قفل ورودی خانه انجام دهد. قفل ورودی دو عدد  $n$  و  $k$  به آلبرت می‌دهد. هر کس که تمام اعداد  $n$  رقمی که اختلاف ارقام متوالی آن دقیقاً  $k$  است را به عنوان پاسخ برای قفل خانه وارد کند، موفق می‌شود وارد خانه شود. اما آلبرت خسته‌تر از آن است که این کار را به تنهایی انجام دهد و از شما می‌خواهد به او کمک کنید تا توجه کنید که سمت چپ‌ترین رقم هر عدد می‌تواند یکی از مقادیر ۱ تا ۹ باشد. همچنین در این سوال استفاده از بردار دو بعدی مجاز نیست.

### ورودی

در تنها خط ورودی، به ترتیب دو عدد  $n$  و سپس  $k$  داده می‌شود.

### خروجی

هر کدام از اعداد مورد نظر را در یک سطر به ترتیب صعودی نمایش دهید.

### محدودیت‌ها

- $1 \leq n \leq 9$
- $0 \leq k \leq 9$

### ورودی و خروجی نمونه

ورودی	خروجی
3 7	181 292 707 818 929

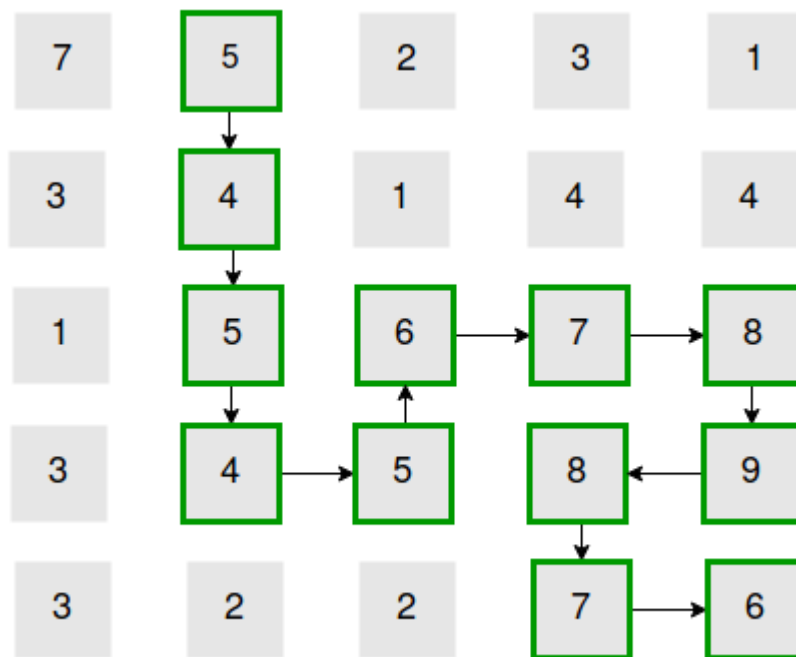
### شرح ورودی و خروجی نمونه

تمام اعداد سه رقمی که قدر مطلق اختلاف هر دو رقمشان 7 باشد اعداد 181، 292، 707، 818، 929 هستند

### ۳. پارکور میان نردها

#### شرح مسئله

پس از اینکه آلبرت با کمک شما موفق شد به خانه برسد، طی اقدامی عجیب تصمیم گرفت رکورد پارکور در شهر نردها را بشکند. برای این کار تصمیم گرفت که بین پشت‌بام‌های خانه‌های شهرشان بپرد. توانایی پرش او در حدی است که تنها می‌تواند بین خانه‌های مجاور (بالا، پایین، چپ و راست) که اختلاف ارتفاع آنها دقیقاً یک واحد است، بپرد. او به شما نقشه‌ای هوایی از شهر نردها داده و از شما خواسته که به او در این رکوردشکنی کمک کنید!



نمونه ای از نقشه هوایی در شکل بالا آمده است (عددی که در هر مربع نوشته شده، ارتفاع آن خانه است). مسیری که با سبز مشخص شده است، بلندترین مسیر قابل پرش برای آلبرت را نشان می‌دهد. آلبرت از هر خانه‌ای می‌تواند شروع به پرش کند اما نمی‌تواند دو بار به یک خانه پرش کند. از آنجایی که آلبرت تجربه قبلی در این ورزش ندارد، ابتدا از شما می‌خواهد که با داشتن این نقشه، در ۲ مرحله به شکل زیر به او کمک کنید مسیر قابل پرشی پیدا کند.

۳-۱ مسیری به طول دقیقاً ۳ که قابل پرش است را پیدا کنید.

۳-۲ طولانی‌ترین مسیر قابل پرش را به او نشان دهید تا بتواند رکورد بزند.

#### ورودی

در اولین خط ورودی دو عدد  $n$  و  $m$  که به ترتیب طول و عرض شهر نردهاست، داده می‌شود ( $n$  تعداد ستون‌ها و  $m$  تعداد سطرها است).

سپس در  $m$  خط بعدی، در هر خط  $n$  عدد  $h_i$  داده می شود که ارتفاع خانه ها را مشخص میکند.

## خروجی

برای بخش ۱-۳ ارتفاع خانه هایی که در یک مسیر به طول ۳ مجاز دیده می شود را نمایش دهید.  
 برای بخش ۲-۳ ارتفاع خانه هایی که در طولانی ترین مسیر قابل پرش مجاز دیده می شود را نمایش دهید.  
 در هر دو قسمت، در صورتی که مسیر مورد نظر موجود نبود، ۱- را نمایش دهید. همچنین در صورتی که چند مسیر با شرایط ذکر شده وجود داشت یکی از مسیرها را به دلخواه چاپ کنید.  
 دقت کنید که در خروجی، باید ارتفاع خانه های مسیر قابل پرش نمایش داده شود.

## محدودیت ها

- $1 \leq n, m \leq 10^3$

- $1 \leq h[i] \leq 10^9$

## ورودی و خروجی نمونه

قسمت اول مسئله:

ورودی	خروجی
2 3 1 2 3 4 5 6	-1
3 3 1 2 3 3 3 5 1 4 2	2 3 4

قسمت دوم مسئله:

ورودی	خروجی
3 3 1 2 7 3 3 6 1 4 5	1 2 3 4 5 6 7
5 5 7 5 2 3 1 3 4 1 4 4 1 5 6 7 8 3 4 5 8 9 3 2 2 7 6	5 4 5 4 5 6 7 8 9 8 7 6

## نحوه تحویل

- کد مربوط به هر پرسش را به صورت یک پرونده<sup>1</sup> جداگانه و به ترتیب سوالات با نام‌های P1.cpp و P2.cpp و P3-1.cpp و P3-2.cpp بنویسید و سپس همه آن‌ها را به صورت یک پرونده فشرده با نام A2-SID.zip در صفحه eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره دانشجویی شما ۸۱۰۱۹۹۹۹۹ باشد، نام پرونده شما باید A2-810199999.zip باشد که شامل پرونده‌های کدهای شما است.
- استفاده از روش‌های بازگشتی، تمیزی کد، شکستن مرحله به مرحله مسئله و طراحی مناسب در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید. مطلبی در راستای آشنایی بیشتر شما با همین موضوع در صفحه‌ی درس بارگذاری می‌شود.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

## نکات پایانی

- توجه کنید که پرسش‌ها حتماً باید به روش بازگشتی حل شوند، هر چند ممکن است روش‌های دیگری نیز برای حل آن‌ها وجود داشته باشد. در سوال یک و دو استفاده از حلقه مجاز نیست و در سوال ۳ قسمت اصلی سوال (پیدا کردن مسیر پرش) حتماً باید به صورت بازگشتی پیاده‌سازی شود.
- هر سوالی درباره‌ی صورت تمرین دارید را در فروم مربوط به تمرین در صفحه‌ی eLearn درس پیرسید تا از سوال‌های تکراری جلوگیری شود. در صورتی که سوالتان قابل طرح در فروم نبود، در ایمیلی به هر چهار دستیار آموزشی (در قسمت گیرنده، ایمیل هر چهار دستیار آموزشی را قرار دهید) سوالتان را مطرح کنید تا در کمترین زمان پاسخ بگیرد.
- یک نمونه ورودی و خروجی برای هر سوال در قالب یک فایل زیپ از طریق صفحه‌ی eLearn درس در اختیار شما قرار داده می‌شود که می‌توانید برای اطمینان از درستی برنامه‌ی خود از آن‌ها استفاده کنید.

---

<sup>1</sup> File