برنامه‌سازی پیشرفته

تمرین کامپیوتری شماره ۲

مدرس: **رامتین خسروی**

طراحان: **سروش میرزاسروری**، **میثاق محقق**، **نسا عباسی مقدم**، **پاشا براهیمی**

مهلت تحویل: جمعه ۱۲ اسفند ۱۴۰۱، ساعت ۲۳:۵۵

مقدمه

این تمرین برای آشنایی با برنامه‌نویسی بازگشتی طراحی و در قالب چهار 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.

تمرین‌ها

تمرین صفر. دستگرمی

برنامه‌ای بنویسید که هر خط از ورودی را توسط یک تابع بازگشتی پردازش کند و حروف کوچک آن را به حروف بزرگ تبدیل کند. سایر کاراکترهای خط بدون تبدیل باقی می‌مانند. به این ترتیب تابع main برنامه شما می‌تواند چیزی شبیه به این باشد:

```
int main() {
	string line;
	while (getline(cin, line)) {
		cout << to_upper(line) << endl;
	}
	return 0;
}
```

ورودی و خروجی نمونه

ورودی	خروجی
"Productivity jumped as offices closed, and stayed high through 2021."	"PRODUCTIVITY JUMPED AS OFFICES CLOSED, AND STAYED HIGH THROUGH 2021."

تمرین ۰۱. پیمایش میثاقی

شرح مسئله

میثاق به تازگی مینیمم و ماکسیمم‌گیری را یاد گرفته است، به همین علت معلم به او تکلیفی داده که در آن باید یک آرایه n عضوی را طی کند.

پیمایش به این صورت است که در هر مرحله، ابتدا باید عضو i ام آرایه را چاپ کرده و سپس به اندازه j از مکانی که در آن قرار داریم (یعنی همان خانه i)، جلو برویم. برای بدست آوردن مقدار j باید بین مقدار آخرین خانه ای در آن قرار داشتیم و عضو i ام آرایه (محل کنونی پیمایش)، به صورت یکی در میان ماکسیمم و یا مینیمم بگیریم (برای فهم بیشتر بخش ورودی و خروجی نمونه را ببینید).

معلم برای کمک به دانش آموزان به آنها این اطمینان را داده است که هر بار برای شروع محاسبه j، بین دو عدد باید ماکسیمم گرفته شود. از آنجا که در ابتدای پیمایش، خانه قبلی ای نداشتیم، مقدار آن را 0 در نظر میگیریم. در ضمن همیشه حرکت را از ابتدای آرایه شروع می‌کنیم.

طی کردن آرایه تا جایی که بتوان از نقطه ای که در آن قرار داریم جلوتر رفت، ادامه دارد و در غیر اینصورت برنامه پایان می‌یابد.

از آنجایی که این درس برای میثاق جدید است، از شما می‌خواهد که با نوشتن یک تابع بازگشتی به او کمک کنید تا مسئله را حل کند.

راهنمایی: برای این که بدانیم در هر مرحله باید مینیمم یا ماکسیمم را حساب کنیم، می‌توانید از یک پارامتر اضافه برای تابع استفاده کنید.

قالب ورودی

در خط اول ورودی عدد n داده می‌شود و در خط دوم n عدد حسابی که با یک فاصله از هم جدا شده‌اند به برنامه داده می‌شود.

قالب خروجی

خروجی باید شامل دنباله‌ای از اعضای آرایه باشد که طبق صورت سوال چاپ می‌شوند.

ورودی و خروجی نمونه

ورودی	خروجی
12 2 5 3 1 0 4 6 1 2 4 3 5	2 3 0 1 1 2 4

توضیحات:

- در مرحله اول، عضو شماره 0 آرایه (عدد 2) چاپ می‌شود و به اندازه $max(0,\ 2) = 2$ جلو می‌رویم.
- در مرحله دوم، عضو شماره 2 آرایه (عدد 3) چاپ می‌شود و به اندازه $min(2,\ 3) = 2$ جلو می‌رویم.
- در مرحله سوم، عضو شماره 4 آرایه (عدد 0) چاپ می‌شود و به اندازه $max(3,\ 0) = 3$ جلو می‌رویم.
- در مرحله چهارم، عضو شماره 7 آرایه (عدد 1) چاپ می‌شود و به اندازه $min(0,\ 1) = 0$ جلو می‌رویم.
- در مرحله بعد، دوباره عضو شماره 7 آرایه (عدد 1) چاپ می‌شود و به اندازه $max(1,\ 1) = 1$ جلو می‌رویم.
- در مرحله ششم، عضو شماره 8 آرایه (عدد 2) چاپ می‌شود و به اندازه $min(1,\ 2) = 1$ جلو می‌رویم.
- در مرحله هفتم، عضو شماره 9 آرایه (عدد 4) چاپ می‌شود و به اندازه $max(2,\ 4) = 4$ جلو می‌رویم.

اکنون متوجه می‌شویم که به اندیس 13 رسیده‌ایم که خارج از محدوده آرایه است. در نتیجه، برنامه به پایان می‌رسد.

تمرین ۲. سایمون بازیگوش

شرح مسئله

سایمون که موشی کنجکاو است می‌خواهد تمامی مسیرهایی که به قالب پنیر می‌رسد را پیدا کند. مسیر، یک ماتریس $n \times n$ است که سایمون در بالا سمت چپ آن قرار دارد و باید به قالب پنیر که در قسمت پایین سمت راست قرار دارد، برسد اما محدودیت‌هایی برای گذر از این مسیرها برای او وجود دارد. سایمون فقط می‌تواند روی ضلع‌های جدول حرکت کند، و جهت حرکت آن تنها به سمت راست و پایین است. همچنین باید دقت کند که از قطر اصلی جدول بالاتر نرود. از آنجایی که سایمون بسیار بازیگوش است، هربار پس از گذشت چند مرحله، حواسش پرت می‌شود و باید دوباره از اول شمارش مسیرها را شروع کند؛ به همین دلیل از شما می‌خواهد که به او کمک کنید تا مسئله را حل کند.

قالب ورودی

ورودی شامل یک خط است که مقدار n در آن داده شده است.

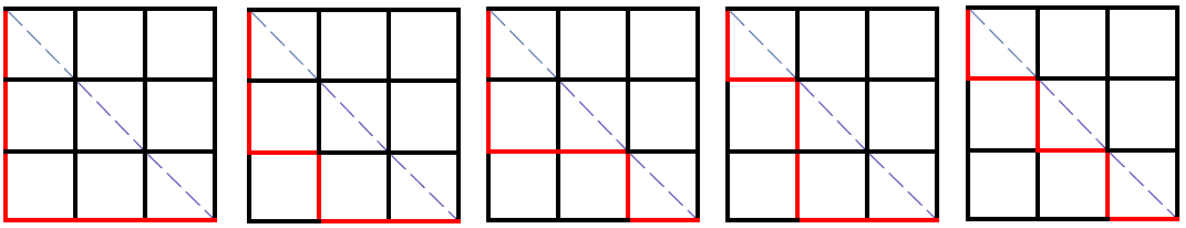
قالب خروجی

خروجی شامل یک عدد است که تعداد تمامی مسیرهایی که میتوان با گذر از محدودیت‌ها از نقطه شروع به نقطه پایان رسید را نشان می‌دهد.

ورودی و خروجی نمونه

ورودی	خروجی
3	5

توضیحات:



(۱) (۲) (۳) (۴) (۵)

تمرین ۰۳. بازی

شرح مسئله

سروش که حوصله‌اش از بازی‌های کامپیوتری سر رفته است، یک بازی مشابه ریورسی[1] اختراع کرده است. صفحه این بازی از یک مربع 7x7 تشکیل شده است؛ امّا فقط 33 خانه خالی در این صفحه وجود دارد. برای اینکه بهتر بتوانید صفحه بازی را تصور کنید، تصویر زیر را مشاهده کنید:

X X O O O X X
X X O O O X X
O O O O O O O
O O O O O O O
O O O O O O O
X X O O O X X
X X O O O X X

در این تصویر، در بخش‌هایی که حرف O قرار گرفته یک خانه خالی وجود دارد، و در بخش‌هایی که حرف X قرار گرفته، هیچ خانه خالی‌ای وجود ندارد و این بخش‌ها در بازی مشارکت نخواهند داشت. زمانی که بازی آغاز می‌شود، تعداد 32 مهره در صفحه بازی وجود دارند به طوری که فقط خانه وسط صفحه خالی است (مهره‌ها را با حرف N نشان می‌دهیم). برای درک بهتر، تصویر زیر را مشاهده کنید:

X X N N N X X
X X N N N X X
N N N N N N N
N N N O N N N
N N N N N N N
X X N N N X X
X X N N N X X

بازی به این صورت است که در هر مرحله، یک مهره می‌تواند **دقیقا** از روی یک مهره مجاورش بپرد به طوری که روی یک خانه خالی فرود آید. بدیهی‌ست که فقط مهره‌هایی را می‌توانیم حرکت دهیم که در فاصله دو واحدی از یک خانه خالی قرار داشته باشند و بین آن مهره و خانه خالی نیز یک مهره وجود داشته باشد. پس از انجام این حرکت، مهره‌ای که از روی آن پرش انجام شده است، از بازی حذف شده و خانه مربوط به آن مهره خالی می‌شود. در نهایت اگر فقط یک مهره در صفحه بماند و آن مهره هم در خانه وسط صفحه باشد، برنده بازی خواهیم بود و اگر بیشتر از یک مهره باقی بماند امّا دیگر نتوانیم پرشی انجام دهیم، یا اینکه فقط یک مهره در صفحه بماند اما در خانه وسط صفحه نباشد، بازنده خواهیم بود.

یکی از دوستان سروش که از این بازی خوشش آمده است، می‌خواهد همیشه برنده این بازی باشد امّا هربار در میانه‌ی بازی ناامید می‌شود؛ به همین دلیل از شما خواسته است که برنامه‌ای طراحی کنید که صفحه بازی را به عنوان ورودی می‌گیرد و

[1] Reversi

مراحل مورد نیاز برای برنده شدن در بازی را در خروجی چاپ می‌کند. لازم به ذکر است که برنامه شما باید حتما با استفاده از روش backtracking مسئله را حل کند.

قالب ورودی

ورودی برنامه در 7 خط خواهد بود که هر خط شامل 7 حرف است:

- حرف X به معنای این است که خانه متناظر آن حرف بخشی از صفحه بازی نیست. دقت کنید که جای حروف X در تمام ورودی‌ها ثابت و مطابق توضیحات ذکر شده است.
- حرف N به معنی وجود مهره در آن خانه است.
- حرف O به این معنی است که خانه متناظر آن حرف در صفحه خالی است.

دقت کنید که در ورودی هیچ فاصله‌ای بین حروف یک خط وجود ندارد.

قالب خروجی

در خروجی باید مراحل مورد نیاز برای برنده شدن در بازی را چاپ کنید. لازم به ذکر است که این مراحل لزوما یکتا نیستند و تا زمانی که برنامه شما یک خروجی صحیح تولید کند، نمره بخش تست‌های این سوال را خواهید گرفت.

برای چاپ مراحل، ابتدا به تصویر زیر که نام هر خانه صفحه را مشخص می‌کند توجه کنید:

A1 A2 A3 A4 A5 A6 A7
B1 B2 B3 B4 B5 B6 B7
C1 C2 C3 C4 C5 C6 C7
D1 D2 D3 D4 D5 D6 D7
E1 E2 E3 E4 E5 E6 E7
F1 F2 F3 F4 F5 F6 F7
G1 G2 G3 G4 G5 G6 G7

فرض کنید که یک مهره در خانه سوم از سمت چپ و در ردیف اول قرار دارد (خانه A3) و خانه C3 هم خالی است. در این‌صورت اگر بخواهیم مهره‌ای که در خانه A3 قرار دارد از روی مهره‌ای که در خانه B3 قرار دارد بپرد و وارد خانه C3 شود، مرحله را به این‌صورت در خروجی نمایش می‌دهیم:

A3 DOWN

هر مرحله شامل دو بخش است: خانه مبدا مهره‌ای که قرار است پرش را انجام دهد و جهت این حرکت که می‌تواند RIGHT یا LEFT یا DOWN و یا UP باشد.

در صورتی که امکان برنده شدن در بازی وجود نداشت، عبارت Loser را در خروجی چاپ کنید.

ورودی و خروجی نمونه

ورودی	خروجی
XXOOOXX XXOOOXX OOOOOOO OONOONO OOOONNO XXONOXX XXOOOXX	E6 LEFT F4 UP D3 RIGHT D6 LEFT

توضیح: پس از انجام مراحل ذکر شده در خروجی، صفحه به شکل زیر خواهد بود:

X X O O O X X
X X O O O X X
O O O O O O O
O O O N O O O
O O O O O O O
X X O O O X X
X X O O O X X

همان‌طور که پیش‌تر ذکر شد، هدف بازی این است که فقط یک مهره در صفحه بماند و آن مهره در خانه وسط صفحه باشد.

ورودی	خروجی
XXOOOXX XXOOOXX OONOOOO OONOOOO OOOOOOO XXONOXX XXOOOXX	Loser

توضیح: با این چیدمان نمی‌توانیم برنده بازی باشیم و در نتیجه عبارت Loser در خروجی چاپ می‌شود.

نکات و نحوهٔ تحویل

- کد هر سوال را در یک فایل مجزا با فرمت Q#.cpp قرار دهید. برای مثال نام فایل حاوی کد پاسخ سوال 1 می‌شود Q1.cpp. سپس کدهای خود را در قالب یک فایل فشرده با نام A2-SID.zip در صفحه‌ی ایلرن درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۱۰۰۰ باشد، نام پروندهٔ کد شما باید A2-810101000.zip باشد که شامل کد شما است.

- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی های آزمون اجرا شود.

- در این تمرین نیز مانند دیگر تمرین‌ها تمیزی کد، شکستن مرحله‌به‌مرحلهٔ مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمرهٔ شما را تعیین خواهد کرد.

- درستی برنامهٔ شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید.

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.