

تمرین سری چهارم

مبانی برنامه‌نویسی کامپیوتر

دکتر ملکی مجد

۲۸ آذر ۱۴۰۳

پاییز ۱۴۰۳



دانشگاه علم و صنعت ایران

۱ مراحل صعود:

تصور کنید که در حال بالا رفتن از یک راه پله با n پله هستید و می‌خواهید به بالای آن برسید. شما می‌توانید هر بار ۱ پله یا ۲ پله بردارید. هدف این است که تعداد روش‌های متمایزی که می‌توانید به بالای پله‌ها برسید را محاسبه کنید.

توجه: برای پاسخ به این سوال باید از یک تابع بازگشتی استفاده کنید!

۱.۱ مثال‌ها:

۱.۱.۱ ورودی:

۲

۲.۱.۱ خروجی:

۲

توضیح:

• روش ۱: ۱ پله + ۱ پله

• روش ۲: ۲ پله

۳.۱.۱ ورودی:

۳

۴.۱.۱ خروجی:

۳

توضیح:

• روش ۱: ۱ پله + ۱ پله + ۱ پله

• روش ۲: ۱ پله + ۲ پله

• روش ۳: ۲ پله + ۱ پله

۲.۱ محدودیت‌ها:

$$1 \leq n \leq 45$$

۲ تعداد ترکیب‌های پرانتز:

به شما یک عدد صحیح n داده شده است که نشان‌دهنده تعداد جفت پرانتزها می‌باشد. وظیفه شما این است که تعداد کل ترکیب‌های معتبر از پرانتزهای به‌درستی فرم‌یافته را شمارش کنید.

توجه: برای پاسخ به این سوال باید از یک تابع بازگشتی استفاده کنید!

۱.۲ ورودی:

یک عدد صحیح n که نشان‌دهنده تعداد جفت پرانتزها است.

۲.۲ خروجی:

یک عدد صحیح که نشان‌دهنده تعداد ترکیب‌های معتبر از پرانتزهای به‌درستی فرم‌یافته است.

۳.۲ مثال‌ها:

۱.۳.۲ ورودی:

۳

۲.۳.۲ خروجی:

۵

ترکیب‌های معتبر به شرح زیر هستند:

((()))
(()())
(())()
()(())
()()()

۴.۲ محدودیت‌ها:

$$1 \leq n \leq 17$$

۳ شبیه‌سازی چراغ‌های ترافیکی:

وظیفه شما شبیه‌سازی چراغ‌های ترافیکی در یک تقاطع شهری با سه خط ترافیکی است. هر خط ترافیکی چراغ خود را دارد که یک چرخه سه‌فازی را دنبال می‌کند:

- چراغ قرمز (با مدت زمان مشخص توسط کاربر)،
- چراغ سبز (با مدت زمان مشخص توسط کاربر)،
- چراغ زرد (مدت زمان ثابت ۱ ثانیه).

برای هر خط، مدت زمان چراغ قرمز و سبز به عنوان ورودی داده می شود. چرخه چراغ برای هر خط به شرح زیر است:

- مدت زمان چراغ قرمز توسط ورودی کاربر برای هر خط تعریف می شود.
 - پس از چراغ قرمز، چراغ ترافیکی به مدت زمان مشخص شده سبز می شود.
 - پس از چراغ سبز، چراغ به مدت ۱ ثانیه زرد می شود.
 - پس از اتمام چراغ زرد، چرخه دوباره با چراغ قرمز آغاز می شود.
- شبیه سازی باید وضعیت (قرمز، سبز یا زرد) هر خط را برای هر ثانیه از ۱ تا n چاپ کند، جایی که n تعداد کل ثانیه های شبیه سازی است.

۱.۳ ورودی:

یک عدد صحیح بین ۱ تا ۱۰۰ که نشان دهنده تعداد کل ثانیه های شبیه سازی است.

$$1 \leq n \leq 100$$

سه عدد صحیح بین ۱ تا ۲۰ که نشان دهنده مدت زمان چراغ سبز (بر حسب ثانیه) برای خط ۱، خط ۲ و خط ۳ هستند.

$$1 \leq x_1, x_2, x_3 \leq 20$$

سه عدد صحیح بین ۱ تا ۲۰ که نشان دهنده مدت زمان چراغ قرمز (بر حسب ثانیه) برای خط ۱، خط ۲ و خط ۳ هستند.

$$1 \leq r_1, r_2, r_3 \leq 20$$

۲.۳ خروجی:

برای هر ثانیه از ۱ تا n ، وضعیت هر خط را به فرمت زیر چاپ کنید:

Second [time]: Lane 1: [state] | Lane 2: [state] | Lane 3: [state]

که در آن [state] می تواند یکی از موارد زیر باشد:

- **قرمز:** چراغ قرمز است.
- **سبز:** چراغ سبز است.
- **زرد:** چراغ زرد است.

۳.۳ نکات

- از یک enum برای نمایش وضعیت‌های چراغ ترافیکی (قرمز، سبز و زرد) استفاده کنید.
- برای هر ثانیه، برنامه باید وضعیت چراغ ترافیکی هر خط را تعیین کرده و آن را به فرمت مورد نظر چاپ کند.
- از enum برای مدیریت وضعیت‌های چراغ‌ها استفاده کنید تا کد شما قابل خواندن‌تر و قابل نگهداری‌تر باشد.
- هر خط چراغ چرخه خاص خود را دارد که بر اساس مدت‌زمان‌های ورودی قرمز و سبز تعیین می‌شود.
- شمارش ثانیه از ۰ شروع می‌شود و تا $n - 1$ ادامه می‌یابد.
- اگر جفت زمان قرمز و سبز صفر باشد، چراغ مدام زرد خواهد بود.

۴.۳ مثال‌ها:

۱.۴.۳ ورودی:

```
10
3 5 7
4 6 3
```

این یعنی شبیه‌سازی برای ۱۰ ثانیه اجرا می‌شود.

- خط ۱ چراغ سبز به مدت ۳ ثانیه و چراغ قرمز به مدت ۴ ثانیه دارد.
- خط ۲ چراغ سبز به مدت ۵ ثانیه و چراغ قرمز به مدت ۶ ثانیه دارد.
- خط ۳ چراغ سبز به مدت ۷ ثانیه و چراغ قرمز به مدت ۳ ثانیه دارد.

۲.۴.۳ خروجی:

```
Second 0: Lane 1: Red | Lane 2: Red | Lane 3: Red
Second 1: Lane 1: Red | Lane 2: Red | Lane 3: Red
Second 2: Lane 1: Red | Lane 2: Red | Lane 3: Red
Second 3: Lane 1: Red | Lane 2: Red | Lane 3: Green
Second 4: Lane 1: Green | Lane 2: Red | Lane 3: Green
Second 5: Lane 1: Green | Lane 2: Red | Lane 3: Green
Second 6: Lane 1: Green | Lane 2: Green | Lane 3: Green
Second 7: Lane 1: Yellow | Lane 2: Green | Lane 3: Green
Second 8: Lane 1: Red | Lane 2: Green | Lane 3: Green
Second 9: Lane 1: Red | Lane 2: Green | Lane 3: Green
```

۴ تعداد راه‌های رمزگشایی پیام:

شما یک پیام مخفی که به صورت یک رشته از اعداد رمزگذاری شده است را دریافت کرده‌اید. این پیام از طریق نگاشت زیر رمزگشایی می‌شود:

"1" -> 'A'
"2" -> 'B'
...
"25" -> 'Y'
"26" -> 'Z'

با این حال، هنگام رمزگشایی پیام، متوجه می‌شوید که راه‌های مختلفی برای رمزگشایی پیام وجود دارد، زیرا بعضی از کدها در کدهای دیگر گنجانده شده‌اند ("۲" و "۵" در مقابل "۲۵").

برای مثال، "۱۱۰۶" می‌تواند به روش‌های زیر رمزگشایی شود:

• "AAJF" با گروه‌بندی (۱، ۱۰، ۱، ۶)

• "KJF" با گروه‌بندی (۱۱، ۱۰، ۶)

گروه‌بندی (۱، ۱۱، ۰۶) نامعتبر است زیرا "۰۶" کد معتبر نیست (فقط "۶" معتبر است).

توجه: ممکن است رشته‌هایی وجود داشته باشند که به هیچ وجه نمی‌توان آن‌ها را رمزگشایی کرد.

با توجه به یک رشته s که فقط از اعداد تشکیل شده است، تعداد راه‌های ممکن برای رمزگشایی آن را بازگردانید. اگر رشته نمی‌تواند به هیچ روش معتبری رمزگشایی شود، ۰ را بازگردانید.

۱.۴ نکته

در این سوال در صورت نیاز می‌توانید از آرایه و رشته استفاده کنید.

۱.۱.۴ ورودی:

رشته‌ای از اعداد s که فقط از ارقام تشکیل شده است.

۲.۱.۴ خروجی:

یک عدد صحیح که نشان‌دهنده تعداد راه‌های معتبر برای رمزگشایی رشته است.

۳.۱.۴ مثال‌ها:

۱.۳.۱.۴ ورودی: "۱۲"

۲.۳.۱.۴ خروجی: ۲

توضیح: "۱۲" می‌تواند به دو روش رمزگشایی شود:

- "AB" (1 2)

- "L" (12)

۳.۳.۱.۴ ورودی: "۲۲۶"

۴.۳.۱.۴ خروجی: ۳

توضیح: "۲۲۶" می‌تواند به سه روش رمزگشایی شود:

- "BZ" (2 26)

- "VF" (22 6)

- "BBF" (2 2 6)

۵.۳.۱.۴ ورودی: "۰۶"

۶.۳.۱.۴ خروجی: ۰

توضیح: "۰۶" نمی‌تواند به "F" رمزگشایی شود زیرا به دلیل صفر پیشرو ("۶" با "۰۶" متفاوت است). در این صورت، رشته یک رمزگذاری معتبر نیست و باید ۰ را بازگردانید.

۴.۱.۴ محدودیت‌ها:

`1 <= s.length <= 100`

s فقط شامل ارقام است و ممکن است شامل صفرهای پیشرو باشد.

۵ ماجراجویی در جنگل

یک برنامه بنویسید که شبیه‌سازی **ماجراجویی در جنگل** را انجام دهد. در این بازی، بازیکن با چالش‌های مختلفی روبرو می‌شود که هر نوع چالش با استفاده از یک **enum** نمایش داده می‌شود. چالش‌ها شامل جمع‌آوری گنج، مبارزه با هیولاها، حل معماها، و استراحت هستند.

برنامه باید با استفاده از یک **تابع بازگشتی**، پیشرفت بازیکن را در جنگل تا زمانی که:

۱. بازیکن با موفقیت از جنگل خارج شود، یا

۲. انرژی او تمام شود،

ردیابی کند.

۱.۵ جزئیات مسئله:

۱.۱.۵ تعریف enum برای چالش‌ها

یک enum به نام Challenge تعریف کنید که شامل موارد زیر باشد:

• Treasure گنج

• Monster هیولا

• Puzzle معما

• Rest استراحت

۲.۱.۵ تولید تصادفی چالش‌ها

جنگل به صورت مرحله‌ای تقسیم شده است و هر مرحله شامل یک چالش تصادفی است که با استفاده از تابع rand() تعیین می‌شود. برنامه باید چالش‌ها را به صورت تصادفی تولید کند.

۳.۱.۵ وضعیت بازیکن

بازیکن با شرایط زیر بازی را شروع می‌کند:

• انرژی: ۱۰۰

• طلا: ۰

۴.۱.۵ نحوه برخورد با چالش‌ها

تابع بازگشتی navigateForest() چالش‌ها را به شرح زیر مدیریت می‌کند:

• **گنج:** مقدار تصادفی طلا (بین ۱۰ تا ۵۰) به کل طلای بازیکن اضافه می‌شود. انرژی بازیکن ۱۰ واحد کاهش می‌یابد. اگر بازیکن انرژی کمتر از ۱۰ داشته باشد، گنج را جمع‌آوری نمی‌کند و انرژی بازیکن صفر می‌شود.

• **هیولا:** بازیکن با یک هیولا مبارزه می‌کند که انرژی او را ۳۰ واحد کاهش می‌دهد. اگر انرژی بازیکن کمتر از ۳۰ باشد، بازی را می‌بازد.

• **معما:** حل یک معما ۲۰ واحد از انرژی بازیکن کم می‌کند. اگر بازیکن انرژی کافی برای حل معما نداشته باشد، بازی را می‌بازد.

• **استراحت:** بازیکن استراحت کرده و ۲۰ واحد انرژی بازیابی می‌کند (تا حداکثر ۱۰۰ واحد).

بازی به صورت بازگشتی ادامه می‌یابد تا زمانی که یکی از شرایط زیر برقرار شود:

• انرژی بازیکن به ۰ یا کمتر برسد (بازی تمام شود).

• بازیکن ۱۰ مرحله از جنگل را پشت سر بگذارد (پیروزی).

۵.۱.۵ پیشرفت و خلاصه بازی

• در پایان هر مرحله، نوع چالش، میزان انرژی، و مقدار طلا نمایش داده می‌شود.

• اگر بازیکن با موفقیت از جنگل خارج شود، پیام پیروزی همراه با کل طلای جمع‌آوری شده نمایش داده می‌شود.

• اگر بازیکن ببازد، پیام «Game Over» و تعداد مراحل طی شده نمایش داده می‌شود.

۶.۱.۵ ورودی و بذر تصادفی

برنامه باید از کاربر یک عدد برای تعیین بذر (seed) تولید تصادفی دریافت کند تا نتایج بازی قابل بازتولید باشد.

۲.۵ مثال:

۱.۲.۵ ورودی:

Enter a random seed: 42

۲.۲.۵ خروجی:

```
Level 1: Challenge - PUZZLE
You solved a puzzle! Energy -20
Current Stats: Gold = 0, Energy = 80

Level 2: Challenge - TREASURE
You found gold! Gold +16, Energy -10
Current Stats: Gold = 16, Energy = 70

Level 3: Challenge - MONSTER
You fought a monster! Energy -30
Current Stats: Gold = 16, Energy = 40

Level 4: Challenge - TREASURE
You found gold! Gold +28, Energy -10
Current Stats: Gold = 44, Energy = 30

Level 5: Challenge - MONSTER
You fought a monster! Energy -30
```

Current Stats: Gold = 44, Energy = 0

Game Over! You ran out of energy at level 6.
