

به نام خدا

سؤال اول) کامپایلر

مهدی این ترم درس کامپایلر دارد. برای این درس استاد این درس یک پروژه به این صورت تعریف کرده است که یک متن دریافت کرده و سپس مشخص می کند آیا رشته ورودی یک رشته معتبر است یا خیر. در صورت معتبر نبودن باید مشخص کند کجای متن ورودی مشکل دارد. برای مثال به دو رشته زیر توجه کنید.

یک رشته معتبر :

(())

یک رشته نامعتبر:

(())

در حقیقت این برنامه چک می کند آیا رشته ورودی هر پرانتز، براکت یا همانند این ها، به طور صحیح باز و بسته شده اند یا خیر. برای برنامه علائم زیر تعریف شده است:

()

[]

{ }

< >

(*)

توجه کنید که دو علامت پرانتز و ستاره با یک دیگر یک علامت را شکل می دهند و رشته (*) یک رشته نامعتبر است. حال شما برنامه ای بنویسید که یک رشته ورودی دریافت و بررسی کند آیا رشته ورودی معتبر است یا خیر؟

ورودی

در ورودی چندین خط به شما داده شده می شود که هر خط به طور مجزا ورودی برنامه است. هر خط حداکثر شامل ۳۰۰۰ کاراکتر می باشد.

خروجی

در هر خط خروجی اگه رشته ورودی صحیح باشد عبارت 'YES' و در غیر این صورت باید عبارت 'NO' را چاپ کند. برای درک بهتر سوال به نمونه توجه کنید.

مثال) ورودی

$(*((\{\} < > (())))) *$

خروجی

YES

سؤال دوم) همکلاسیا

شما احتمالا با ساختمان داده های صف و صف اولویت آشنایی دارید و می دونید که چجوری ساخته شده اند. در اینجا شما با ساختمان داده جدیدی آشنا میشد. این صف به این شکل است که افراد درون یک صف ایستادند. وقتی یک نفر جدید وارد صف می شه از اول صف تا آخر صف را بررسی می کنه اگر همکلاسی های اون در صف قرار داشته باشند به آنها ملحق می شه و دقیقا بعد از اونا می ایسته و اگر هیچ همکلاسی ای در صف پیدا نکنه به آخر صف می ره. در انتها وقتی همه ی دانشجوها در صف قرار گرفتن، از اول صف غذاشونو میگیرن و از صف خارج میشن. حال شما این صف را پیاده سازی کنید.

ورودی

ورودی شامل چندین تست کیس می باشد. در خط اول هر تست کیس ، تعداد رشته ها آمده است. ($t < 1001$) در t خط بعدی در ابتدا تعداد دانشجوی هر رشته آمده است و سپس به دنبال آن شماره دانشجویی هر دانشجو. هر رشته حداکثر 1000 دانشجو دارد. شماره های دانشجویی در رنج 0 تا 10000 هستند. در نهایت یک لیست از دستورات آمده است که دستورات به شرح زیر است:

دانشجویی با شماره دانشجویی خود وارد صف می شود. - ENQUEUE x

تحویل غذا به نفر اول صف - DEQUEUE

پایان دستورات - STOP

خروجی

برای هر تست کیس در خط اول عبارت Scenario #k چاپ شود که k شماره تست کیس است. سپس برای هر دستور DEQUEUE شماره دانشجویی که غذا گرفته است را چاپ کنید. بعد از پایان هر تست کیس یک خط خالی چاپ شود.

2

3 101 102 103

3 201 202 203

ENQUEUE 101

ENQUEUE 201

ENQUEUE 102

ENQUEUE 202

ENQUEUE 103

ENQUEUE 203

DEQUEUE

DEQUEUE

DEQUEUE

DEQUEUE

DEQUEUE

DEQUEUE

STOP

Scenario #1

101

102

103

201

202

203

سؤال سوم) کیبورد خراب

کیبورد لپتاپ من خراب شده و هر از گاهی خود به خود دکمه های `home` یا `end` رو میزنه. حالا من از شما میخام ببینید زمان هایی که کیبورد واسه ی خودش دکمه میزنه خروجی چی میشه.

ورودی:

ورودی شامل چندین خط میباشد که شامل دو کاراکتر کلید `[` و `]` میباشد. هنگام درج `[` کیبورد به اول خط میرود و در هنگام درج `]` کیبورد به آخر خط میرود.

خروجی:

شما باید متنی که در نهایت توسط لپتاپ خروجی داده میشود را چاپ کنید.

مثال) ورودی

```
This_is_a_[nice]_text
[[]][[]]Data_Structures
```

خروجی

```
niceThis_is_a__text
Data_Structures
```

سؤال چهارم) BWF

در این تمرین هدف پیاده سازی حافظه و مدیریت آن توسط لیست های پیوندی دوطرفه است. بدین منظور سه استراتژی برای اختصاص حافظه و یک روش آزادسازی می بایست پیاده سازی شوند.

استراتژی های اختصاص حافظه به سه شکل زیر خواهد بود:

- **Best fit**: اختصاص حافظه از نزدیک ترین مقادیر آزاد موجود
- **Worst fit**: اختصاص حافظه مورد نظر از بزرگترین حجم آزاد موجود.
- **First fit**: اختصاص حافظه از اولین حجم آزاد پیدا شده با اندازه ای متناسب برای حجم اختصاصی مورد نظر.

ابتدا یک گره که مبین میزان حجم حافظه (نقطه شروع و پایان حافظه) است، با نام **memory** تولید می شود (نقطه شروع را 1 در نظر بگیرید) سپس با دستور **allocate** و مشخص نمودن حجم حافظه و نام آن بخش از فضا، با استراتژی از پیش مشخص شده فضا را تقسیم می کنیم. با دستور **free** و دادن نام فضا، آن بخش آزاد می شود. توجه داشته باشید که پس از آزادسازی، مجموعه فضاهای آزاد متوالی باید به صورت یک گره یکپارچه شوند.

برنامه باید حجم کل حافظه، استراتژی اختصاص حافظه و مجموعه ای از دستورات را از فایل ورودی **in.txt** دریافت کرده و بخش های آزاد و اشغال شده را نهایتاً در فایل خروجی **out.txt** چاپ نماید.

نمونه ای از فایل ورودی (ستون سمت چپ) و خروجی (ستون سمت راست) در زیر آمده است:

10000	allocated:
first fit	m1(1,500)
allocate(500,m1)	m4(501,900)
allocate(5000,m2)	m3(5501,5700)
allocate(200,m3)	free:

free(m2)	901-5500
allocate(400,m4)	5701-10000