برنامه‌سازی پیشرفته **تمرین کامپیوتری شماره ۲**

مدرس: رامتین خسروی

طراحان: هادی بابالو، علی پادیاو، امیرعلی رحیمی، نسا عباسی، عرفان عسگری، آوا میرمحمدمهدی، مهدی نوری

مهلت تحویل: سه‌شنبه ۲۲ اسفند ۱۴۰۲، ساعت ۲۳:۵۵

مقدمه

این تمرین برای آشنایی با برنامه‌نویسی بازگشتی طراحی و 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. دقت کنید در مسئله آخر صرفا حل کلی مسئله و بک‌ترکینگ آن باید به صورت بازگشتی حل شود و در باقی بخش‌های آن، استفاده از حلقه‌ها مانعی ندارد.

تمرین‌ها

تمرین ۱. ساخت اعداد با رقم ۱

شرح مسئله

از آنجا که سلطان زمان زیادی را به تنهایی در اتاق ACM و در سایت دانشکده برق و کامپیوتر سپری می‌کند، به مسائل عجیب زیادی فکر می‌کند. یکی از مسائلی که جدیدا ذهن سلطان را به خود درگیر کرده است، ساخت اعداد تنها با رقم ۱ است. سلطان می‌خواهد هر عددی که دوست دارد را صرفا با استفاده از رقم ۱ و عملیات‌های پایه جمع و تفریق بسازد. به طور مثال می‌توان عدد ۱۲۱۲ را به صورت ۱۱۱۱+۱۱۱-۱۱+۱ ساخت و برای ساخت آن حداقل به ۱۰ رقم نیاز داریم. حالا سلطان به کمک شما نیاز دارد تا بفهمد که برای ساخت عدد n **حداقل** به چند رقم ۱ نیاز دارد؟

قالب ورودی

ورودی تنها شامل یک خط است که در آن عدد طبیعی n آمده است.

$$1 \leq n < 10^{15}$$

قالب خروجی

در یک خط حداقل تعداد ارقام ۱ را برای ساخت n چاپ کنید.

ورودی و خروجی نمونه

خروجی ۱	ورودی ۱
3	10

برای ساخت عدد ۱۰ حداقل به ۳ رقم نیاز داریم و می‌توانیم آن را به صورت ۱-۱۱ بسازیم.

خروجی ۲	ورودی ۲
4	110

به طور مشابه برای تشکیل ۱۱۰ نیاز به حداقل ۴ رقم داریم و می‌توانیم این کار را به صورت ۱-۱۱۱ انجام دهیم.

خروجی ۳	ورودی ۳
10	1212

تمرین ۲. نمایش ساختار فولدرها

شرح مسئله

سلطان در یکی از پروژه‌هایش قصد پیاده‌سازی ترمینال لینوکس را دارد. در یکی از مراحل سلطان قصد دارد کاربرد tree در لینوکس را پیاده‌سازی کند. این دستور برای نمایش فایل‌ها و پوشه‌های یک دایرکتوری استفاده می‌شود.

برای مشاهده فایل‌ها و پوشه‌های یک دایرکتوری در کنسول که امکانات گرافیکی زیادی در اختیار نیست، این روش استفاده می‌گردد. برای نمایش فهرست درختی یک دایرکتوری، در هر خط نام یک پوشه یا فایل چاپ می‌شود و ارتباط بین آنها با کاراکترهایی که بازنمایی‌کننده شاخه‌های درخت هستند مشخص می‌گردد. در این روش، چنانچه یک فایل یا پوشه درون یک پوشه دیگر قرار داشته باشد به صورت زیرشاخه‌ای از آن پوشه نمایش داده می‌شود.
زمانی که عمق پوشه‌های یک دایرکتوری زیاد باشد، چاپ کردن همه اجزا آن جاگیر است. در چنین شرایطی برای جلوگیری از شلوغ شدن نمایش، چاپ کردن زیرشاخه‌های آن را به عمقی خاص محدود می‌کنند.
دراین تمرین شما باید به سلطان کمک کنید تا با دریافت کردن لیستی از فایل‌ها و پوشه‌های یک دایرکتوری، فهرست درختی کل آن دایرکتوری را تا عمق خاصی چاپ کند.

قالب ورودی

در سطر اول عدد طبیعی n (تعداد آیتم‌ها) و پس از آن، عدد طبیعی d (حداکثر عمق نمایش) داده می‌شود.
سپس در n سطر بعدی اطلاعات آیتم‌ها داده می‌شود. در سطر i-ام ابتدا نام آیتم i و سپس با یک فاصله، شماره سطر پوشه‌ای که در آن قرار دارد مشخص می‌شود. به طور مثال اگر نام آیتم پنجم ورودی، file باشد و پوشه‌ای که این فایل در آن قرار دارد در سطر ۳ معرفی شده باشد، سطر ۵ به این صورت خواهد بود:

```
file 3
```

آیتم‌های درون یک پوشه لزوما پشت سر هم در ورودی داده نمی‌شوند اما تضمین می‌شود که آیتم یک پوشه بعد از تعریف آن پوشه در ورودی ظاهر شود. پوشهٔ ریشه[1] با شماره ۰ مشخص می‌شود.

قالب خروجی

در هر خط از خروجی ممکن است چهار مورد زیر چاپ شود:

۱- تعدادی کاراکتر فاصله (Space)
۲- شاخه‌های باز (`|`) (کاراکتر OR)
۳- سرشاخه (`___`) (دو کاراکتر Underline)
۴- نام آیتم

تو رفتگی[2] هر آیتم، مشخص کننده رابطهٔ یک آیتم با آیتم سطر قبل است. به طور مثال اگر یک آیتم نسبت به سطر قبل خود، تورفتگی یکسانی داشته باشد با آن همسان بوده و هر دو در یک پوشه قرار دارند. اگر تورفتگی‌اش بیشتر از سطر قبل باشد به این معنی است که آن آیتم درون آیتم سطر قبل قرار دارد. ترتیب چاپ کردن آیتم‌های همسان باید به همان صورتی باشد که در ورودی آمده است. همچنین به ادامه‌دار بودن چاپ شاخه‌های تمام نشده دقت کنید.
در حین نمایش عمق به این سه نکته دقت کنید:

1. به ازای هر بار ورود به عمق جدید سه کاراکتر به تورفتگی اضافه می‌شود.

---

[1] root folder
[2] indentation

2. این سه کاراکتر تورفتگی می‌تواند به صورت سه کاراکتر فاصله (•••) یا یک شاخه باز و دو کاراکتر فاصله (•• |) باشد. شاخه‌های باز به این معنی هستند که نمایش محتویات یک پوشه هنوز به اتمام نرسیده است و در خطوط بعدی ادامه خواهد یافت.
3. بعد از چاپ نام آخرین آیتم از یک پوشه و قبل از نمایش محتویات آن آیتم، شاخه مربوط به آن پوشه بسته می‌شود و دیگر ادامه نمی‌یابد.

ورودی و خروجی نمونه

توجه: کاراکتر های (•) صرفا برای شفافیت در تعداد کاراکترهای فاصله به این شکل به نمایش درآمده‌اند و شما در خروجی باید همان کاراکتر فاصله ( ) را چاپ کنید.

خروجی ۱	ورودی ۱
|__file1 |__folder1 •••|__file2 •••|__file3 •••|__folder2 •••|••|__file4 •••|__folder3	7 5 file1 0 folder1 0 file2 2 file3 2 folder2 2 file4 5 folder3 2

file1 و folder1 هر دو در پوشه ریشه (۰) قرار دارند. file2, file3, folder2 و folder3 همگی در folder1 (که در سطر ۲ معرفی شده) قرار دارند. file4 در folder2 (که در سطر ۵ معرفی شده) قرار دارد. همچنین عمیق‌ترین آیتم در این دایرکتوری در عمق ۳ قرار دارد پس محدودیت نمایش عمق ۵ تاثیری در چاپ نداشته است.

خروجی ۲	ورودی ۲
|__file1 |__folder1 •••|__file2 •••|__folder2 •••|••|__folder3 •••|•••••|__folder4 •••|•••••|••|__folder5 •••|•••••|__file4 •••|__folder6 ••••••|__folder7 •••••••••|__file5	12 5 file1 0 folder1 0 file2 2 folder2 2 folder3 4 folder4 5 folder5 6 file3 7 file4 5 folder6 2 folder7 10 file5 11

file3 در عمق ۶ قرار قرار دارد، بنابراین در این نمایش که محدودیت عمق ۵ دارد، از چاپ آن صرف نظر شده است.

راهنمایی

انتخاب مناسب ساختار داده تاثیر به سزایی در سادگی حل این مسئله دارد. برای نگهداری داده‌های ورودی، یک راه این است که آن‌ها را به صورت لیستی از دوتایی‌های نام آیتم و شماره آیتم پدر ذخیره کنید.
برای پیاده‌سازی این سوال می‌توانید یک تابع بازگشتی طراحی کنید که آیتم‌های درون یک آیتم را یافته و چاپ کند و برای چاپ محتوای هر آیتم از آن، به صورت بازگشتی خودش را صدا بزند.
یک راه دیگر برای نگهداری ورودی این است که آن‌ها را در دوتایی‌های نام آیتم و لیستی از شماره آیتم‌های فرزند ذخیره کنید. برای اینکار باید در هنگام گرفتن ورودی، شماره هر خط ورودی را به لیست فرزندان آیتم پدرش اضافه کنید. با این روش در تابع بازگشتی دیگر نیازی به جستجو و یافتن فرزندان یک آیتم ندارید.

توجه کنید که با ورود به هر عمق جدید تورفتگی بیشتر می‌شود و این تورفتگی به کاراکترهای تشکیل‌دهنده تورفتگی قبلی اضافه می‌شود؛ اما کاراکترهای تشکیل دهنده تورفتگی‌های قبلی بدون تغییر باقی می‌ماند. بنابراین می‌توانید کاراکترهای تشکیل دهنده تورفتگی تا اینجای کار را برای چاپ به توابع عمیق‌تر پاس دهید.

دقت کنید که روش‌های ذکر شده تنها روش‌های ممکن برای پیاده‌سازی این مسئله نیستند و پیاده‌سازی با روش‌های صحیح دیگر نیز مانعی ندارد.

تمرین ۳. خطوط مترو

شرح مسئله

به دلیل فقر و وضعیت بد اقتصادی، سلطان می‌خواهد برای دیدار با دوستانش از مترو استفاده کند. خطوط مترو به گونه‌ای طراحی شده‌اند که هر دو خط مترو **حداکثر** در یک ایستگاه تقاطع دارند و همچنین هر ایستگاه محل تقاطع **حداکثر** دو خط مترو است. فاصله هر ایستگاه تا ایستگاه مجاورش ۱ دقیقه و عوض کردن خط، ۲ دقیقه زمان می‌برد. حال سلطان به کمک شما نیاز دارد تا بتواند در کمترین زمان ممکن به مقصدش برسد.

قالب ورودی

در سطر اول ورودی عدد n که نشان دهنده‌ی تعداد خطوط مترو است، آمده است. سپس در 2n سطر بعدی به ازای هر خط مترو از 1 تا n:

- در سطر اول تعداد ایستگاه‌های آن خط مترو مشخص شده است.
- در سطر دوم به ازای هر ایستگاه خط مذکور یک عدد نوشته شده است؛ در صورتی که این عدد برابر با 0 باشد، در آن ایستگاه تقاطعی وجود ندارد و در غیر این صورت شماره خطی که در آن ایستگاه تقاطع دارد نوشته شده است.

در سطر پایانی ابتدا شماره خط مترو و سپس شماره ایستگاه مبدا سلطان و در ادامه‌ی آن شماره خط مترو به همراه شماره ایستگاه مقصد سلطان آورده شده است.

توجه کنید که در ایستگاه مبدا و مقصد داده شده، تقاطعی وجود ندارد.

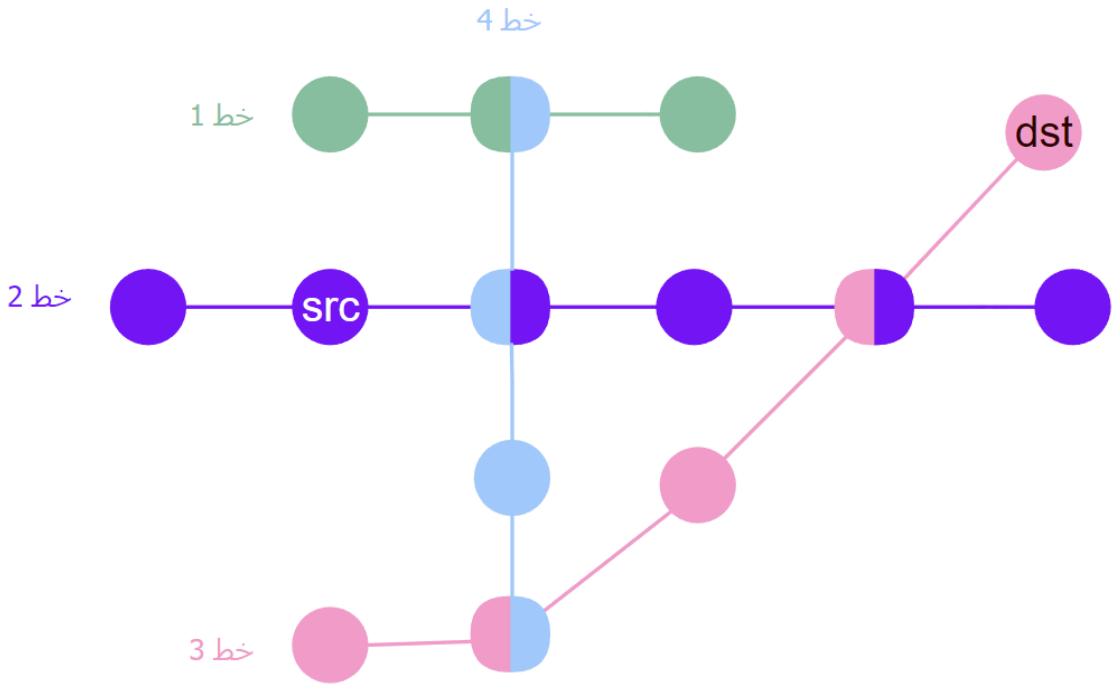
قالب خروجی

در یک خط حداقل زمان لازم برای رسیدن از مبدا به مقصد را چاپ کنید. تضمین می‌شود حداقل یک مسیر از مبدا تا مقصد سلطان وجود داشته باشد.

ورودی و خروجی نمونه

خروجی ۱	ورودی ۱
6	4 3 0 4 0 6 0 0 4 0 3 0 5 0 4 0 2 0 4 1 2 0 3 2 2 3 5

از مبدا تا ایستگاهی که تقاطع خط ۲ و خط ۳ است، ۳ ایستگاه فاصله است که جمعا ۳ دقیقه طول می‌کشد. سپس سلطان باید خطش را عوض کند و به خط ۳ برود که ۲ دقیقه طول می‌کشد و از آنجا به مقصدش در خط ۳ برود که یک دقیقه زمان می‌برد. در نتیجه مجموع کمترین زمان صرف شده ۶ دقیقه است.


4 خط
1 خط
dst
2 خط
src
3 خط

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 ۱ 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 ۸۱۰۱۰۲۰۰۰ باشد، نام پروندهٔ کد شما باید A2-810102000.zip باشد که شامل کد شما است.
- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی های آزمون اجرا شود.
- 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.
- ممکن است نکات جدیدی در خصوص پروژه در تالار گفتگو مطرح شود که در نمره‌دهی و نحوه آزمون پروژه شما موثر خواهد بود.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.

نمرات

- تمیزی کد
  - رعایت کردن نام‌گذاری صحیح و انسجام[3]
  - عدم وجود کد تکراری
  - رعایت دندانه‌گذاری[4]
  - استفاده **صحیح** از متغیرهای ثابت به جای Magic Value-ها
  - ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند
- درستی کد
  - آزمون‌های خودکار
- طراحی بازگشتی
  - پیاده‌سازی الگوریتم بازگشتی و عدم استفاده از حلقه (به جز ورودی/خروجی)

دقت کنید که موارد ذکر شده لزوما کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

---

[3] Consistency

[4] Indentation