



## مقدمه

در این تمرین قرار است کمی با کلیت زبان C++ و کار کردن با آن آشنا شوید.

## ناظر گیت‌هاب

شهریار یک کاربر فعال گیت‌هاب است. او اگر به صفحه گیت‌هابی علاقمند باشد آدرس آن را یادداشت می‌کند تا بعداً دوباره به آن سر بزند. آدرس‌های صفحات گیت‌هاب به صورت زیر است:

`https://<username>.github.io/<context>[/query=<idontknow>]`

- **Username:** نام کاربری صاحب صفحه است که یک رشته شامل حروف کوچک است. این رشته نمی‌تواند خالی باشد.
- **Context:** یک رشته شامل حروف کوچک است. این رشته نمی‌تواند خالی باشد.
- **Idontknow:** یک رشته شامل حروف کوچک است. اگر این رشته خالی باشد، دیگر قسمت مربوط به `"/query="` را نخواهیم داشت.

او آخرین بار آدرسی را با بدون استفاده از کاراکترهای نقطه‌گذاری در دفترچه‌اش نوشت و گمان می‌کرد که آدرس اصلی را بعداً به یاد خواهد آورد. حال شهریار به سراغ دفترچه خود رفته و در پیدا کردن آدرس اصلی صفحات به مشکل خورده است.

شهریار می‌خواهد برنامه‌ای بنویسد که نوشته‌ی او را بگیرد و آدرس اصلی را به او بدهد.

## ورودی

ورودی شامل یک رشته با طول حداکثر 50 کاراکتر است که همان نوشته‌ی شهریار است. تضمین می‌شود ورودی حتماً یک جواب داشته باشد و یک رشته معتبر باشد.

بیکربندی ورودی

`httpseleahexhodaverdigithubiomontecarloquerypi`

# خروجی

خروجی شامل آدرس اصلی نوشته‌ی شهریار است.

پیکربندی خروجی

`https://elahekhodaverdi.github.io/montecarlo/query=pi`

## محاسبات ضعیف (اختیاری)

از آنجا که الهه بسیار سر به هوا است و در محاسبات ساده بسیار بی‌دقتی می‌کند، تصمیم گرفت که یک ماشین‌حساب ساده بنویسد که به او در محاسبات کمک کند. ماشین‌حساب الهه از چهار عمل اصلی ریاضی به همراه عمل به توان رساندن پشتیبانی می‌کند. از آنجا که الهه در کد زدن بسیار بی‌مهارت است از شما کمک خواسته که در نوشتن این برنامه به او کمک کنید.

## ورودی

در خط اول یک عدد  $t$  می‌آید که نشان‌دهنده تعداد عملیاتی است که الهه می‌خواهد با ماشین‌حساب خود حساب کند.

پس از آن در هر خط یک عبارت ریاضی وارد می‌شود (تضمین می‌شود که عبارت‌ها با معنی هستند و نیازی به بررسی این مورد نیست) که حالت کلی آن به شکل زیر می‌باشد:

`digits(space)?operator(space)?digits`

که معادل `regex` آن به شکل زیر می‌باشد:

`\d+\s?[\+\-\/*\^]\s?\d+`

که به این معنی می‌باشد که ابتدا یک یا بیشتر رقم دیده می‌شود (علامت `+` به معنای یکی یا بیشتر است و wildcard از پیش تعریف شده `\d` به معنای یک رقم است)، سپس ممکن است یک فاصله (`space`) دیده شود یا دیده نشود (علامت `?` به این معنی است که ممکن است وجود داشته باشد یا نداشته باشد، توجه داشته باشید که `\s` به طور کلی برای `whitespace`‌ها استفاده می‌شود اما در این سوال صرفاً فاصله ممکن است باشد و حالت دیگری ندارد)، پس از آن یکی از `operator`‌های `+`، `-`، `/`، `*` و `^` دیده می‌شود، و دوباره پس از آن ممکن است فاصله دیده شود یا نه، و در نهایت نیز یک عدد یک یا چند رقمی دوباره خواهیم دید. برای آشنایی بیشتر می‌توانید `regex` بالا در [این سایت](#) وارد کنید و از بخش `explanation` توضیحات مربوطه را بخوانید.

در نهایت شما باید نتیجه محاسبات هر خط را در یک خط جدا نمایش دهید. توجه کنید که ورودی‌ها همه عدد صحیح و مثبت هستند و خروجی شما نیز باید عدد صحیح باشد (برای تفریق ممکن است خروجی منفی باشد)،

برای عمل تقسیم در صورتی که باقی مانده داشت از تابع `floor` می‌توانید استفاده کنید و خروجی صحیح تولید کنید.

پیکربندی ورودی
4
1+2
8^ 3
9 / 4
1 -8

**خروجی**

پیکربندی خروجی
3
512
2
-7

# نکات و نحوه تحویل

- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 یا بالاتر ترجمه و در زمان معقول برای ورودی های آزمون اجرا شود.
- در طول این تمرین ممکن است با مشکلاتی روبه‌رو شوید که راه حل آن‌ها را نمی‌دانید، جست‌وجوگرهایی مانند google و سایت‌هایی مانند [stackoverflow](#) و [cplusplus](#) و [cppreference](#) ممکن است به شما کمک کنند.
- تحویل این تمرین در سامانه کوئرا انجام می‌شود. برای ورود به کلاس در سایت کوئرا می‌توانید از [این لینک](#) استفاده کنید. رمز ورود به کلاس APF03 است. لطفا اسم خود را در کوئرا به فارسی و به صورت کامل ذخیره کنید.
- 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.
- دقت کنید که سوال اول تمرین نمره‌ای ندارد اما انجام آن، اجباری است. سوال دوم تمرین نیز اختیاری است و برای تمرین بیشتر توصیه می‌شود اما انجام دادن آن اجباری نیست.