

# پنج‌شنبه‌ی چهارم

در رقابت امروز نیز اولین جواب درست، بهترین جواب خواهد بود. همچنین، بهتر است نام کاربری، قسمتی از نام و نام خانوادگی شرکت‌کننده را در بر داشته باشد تا بتوان با توجه به آن برنده را تشخیص داد. اعضای تیم فنی رقابت پنج‌شنبه‌های سخت، برای شما بهترین لحظات و پیروزی را در این رقابت آرزو می‌کنند.

پیش‌واکشی تسلنگ	عنوان مسئله
ct04	شناسه‌ی مسئله
۳ از ۹	سختی مسئله
ساعت ۱۶ ۱۳۹۵/۷/۲۲	زمان شروع
ساعت ۱۶ ۱۳۹۵/۷/۲۴	زمان پایان

آقای دال (حرف اول نام یکی از اعضای محترم تیم فنی صد و یک نفره ی پنج‌شنبه‌های سخت) نامه‌ی زیر را از یکی از دوستان قدیمی خود دریافت کرده است.

آقای دال عزیز،

واژه‌ها از توصیف حس غرور تیم بیست و سه نفره‌ی تسلنگ (از این جهت که موضوع پنج‌شنبه‌ی گذشته به ایشان اختصاص داده شده است) ناتوان هستند. این اتفاق که بهترین برنامه‌نویسان جهان برای حل یکی از چالش‌های یک مترجم قدرتمند تلاش کنند، بدون تردید بی‌نظیر است. از طرفی، هدف بزرگ تیم تسلنگ، قوی‌ترین و سریع‌ترین کامپایلر نوشته‌ی بشر است و این هدف بدون چنین کمک‌هایی دست یافتنی نخواهد بود. در هر صورت، اکنون در مترجم تسلنگ با مسئله‌ی با اهمیت دیگری روبرو هستیم. در پردازنده‌ی جدید شرکت آر (یکی از برترین تولیدکننده‌های پردازنده، که پشتیبان مالی مترجم تسلنگ نیز هست) دسترسی به حافظه، تأخیر زیادی دارد (بیش از پنجاه سیکل). بنابراین در صورتی که قسمتی از حافظه خوانده شود و در حافظه‌ی نهان موجود نباشد، پردازنده مدتی بیکار می‌ماند؛ این اتفاق، خطای حافظه‌ی نهان (Cache miss) نامیده می‌شود.

یکی از راه‌های کاهش خطاهای حافظه‌ی نهان، پیش واکشی (Prefetch) نرم‌افزاری است. در این راه، کمی قبل از دسترسی به یک آدرس، از پردازنده درخواست شود که محتوای آن آدرس را در حافظه‌ی نهان قرار دهد. در نتیجه، مترجم تسلنگ می‌تواند چند دستور قبل از دسترسی به آدرسی از حافظه، یک دستور پیش واکشی تولید کند تا در هنگام دسترسی، محتوای آدرس از حافظه‌ی نهان خوانده شود. با توجه به محدودیت‌های سخت‌افزاری، در هر لحظه نباید تعداد زیادی درخواست پیش واکشی در حال انجام باشند. علاوه بر آن، پیش واکشی نباید خیلی دیر انجام شود. هدف ما بدست آوردن حداقل تعداد خطاهای حافظه‌ی نهان در یک بلوک پایه است و مطمئن هستیم تجربه‌ی شما در این مسئله نیز به ما کمک خواهد کرد.

با احترام فراوان،

کاف

خط اول ورودی شامل سه عدد است. عدد اول حداقل تعداد دستوراتی را نشان می‌دهد که باید بین پیش واکشی و دسترسی به حافظه فاصله وجود داشته باشد (حداکثر دویست). عدد دوم حداکثر تعداد درخواست‌های پیش واکشی همزمان (که هنوز دسترسی به آنها انجام نشده است) را نشان می‌دهد (حداکثر صد). عدد سوم تعداد متغیرهای بلوک پایه را نشان می‌دهد (حداکثر پنجاه هزار). سپس به تعداد متغیرها، عدد در ادامه ظاهر می‌شوند: عدد  $i$ -ام شماره‌ی دستوری را نشان می‌دهند که در آن اولین بار متغیر  $i$ -ام از حافظه خوانده می‌شود. شماره‌ی دستورها از صفر شروع می‌شوند و حداکثر صد هزار هستند. خروجی باید حاوی یک عدد باشد که کمینه‌ی تعداد خطاهای حافظه‌ی نهان را نشان می‌دهد.

در مثال زیر، اگر متغیر اول و دوم قبل از دستور دوازدهم پیش واکشی شوند و متغیر ششم قبل از دستور شصتم پیش واکشی شود، تعداد خطاها سه خواهد بود. دقت به این نکته لازم است که چون متغیر سوم در دستور دهم از حافظه خوانده می‌شود و باید بین پیش واکشی و دسترسی حداقل بیست دستور فاصله وجود داشته باشد، پیش واکشی این متغیر ممکن نیست. از طرفی در هر لحظه نمی‌توان بیش از دو پیش واکشی انجام داد، پس پیش واکشی متغیر چهارم و پنجم ممکن نیست. بنابراین، حداقل تعداد خطاها سه می‌باشد.

ورودی	خروجی
20 2 6 32 33 10 30 31 80	3

مثال دیگری با سه متغیر در ادامه نشان داده می‌شود.

ورودی	خروجی
10 1 3 24 29 19	1

برای ثبت نام کاربری و گذرواژه می‌توانید از دستور زیر استفاده کنید که در آن **USERNAME** نام کاربری و **PASSWORD** گذرواژه‌ی پیشنهادی شما هستند.

```
$ echo "register USERNAME PASSWORD" | netcat ct.rudi.ir 40
```

برای ارسال جواب، می‌توانید از دستور زیر استفاده کنید که در آن باید **USERNAME** با نام کاربری، **PASSWORD** با گذرواژه، **LANG** با زبانی که جواب در آن نوشته شده است و **MYPROG** با آدرس فایل جواب جایگزین شوند. به جای عبارت **LANG** مقدار «C»، «C++»، «py» یا «sh» می‌توان قرار داد که به ترتیب نشان می‌دهند جواب با زبان C، C++، Python یا اسکریپت پوسته نوشته شده است.

```
$ (echo "submit USERNAME PASSWORD ct04 LANG"; \
  cat MYPROG; echo EOF) | netcat ct.rudi.ir 40
```

برای مشاهده‌ی نتایج ارزیابی خودکار جواب‌ها، می‌توانید از دستور زیر استفاده کنید.

```
$ echo "report ct04" | netcat ct.rudi.ir 40
```

ستون‌های خروجی به ترتیب نام کاربری ارسال کننده، زمان ارسال، تعداد نمونه‌ها با خروجی درست و مدت اجرا می‌باشند. برای مشاهده‌ی جواب‌های درست ارسال شده، به ترتیب زمان ارسال، می‌توانید از دستور زیر استفاده کنید. جواب اول در خروجی این دستور، بهترین جواب می‌باشد.

```
$ echo "report ct04" | netcat ct.rudi.ir 40 | grep '\.$'
```

در صورت بروز مشکل در اتصال به سرور، در دستورات بالا می‌توانید عبارت «netcat ct.rudi.ir 40» را با عبارت «ssh ct@ct.rudi.ir» جایگزین کنید و وقتی کلمه‌ی عبور از شما درخواست شد، «ct» را وارد نمایید.

همه‌ی افرادی که در این رقابت شرکت می‌کنند باید قوانین زیر را رعایت نمایند.

- الف برای حل هر نمونه، به هر جواب دو ثانیه زمان و پانصد مگابایت حافظه اختصاص می‌یابد.
- ب شرکت‌گندگان آزادند در مورد مسئله‌ها مشورت کنند ولی همه‌ی قسمت‌های جوابی که ارسال می‌شود باید نوشته‌ی شخص ارسال‌کننده باشد.
- ج ارسال دو جواب مشابه توسط دو کاربر مختلف تخلف محسوب می‌شود.
- د بین ارسال جواب‌های یک کاربر باید حداقل دو دقیقه فاصله باشد؛ در غیر این صورت، برنامه‌ی داور خطا گزارش می‌دهد.
- ه استفاده از توازی در جواب‌ها آزاد است؛ می‌توانید از چند پردازنده یا ریسمان استفاده نمایید.
- و در جلسه‌ای با حضور اعضای محترم تیم فنی، دکتر عمران پور و دکتر غلامی، با توجه به موضوع و سختی مسئله، برای درس‌های مرتبط نمره‌ی اضافی در نظر گرفته شده است.
- ز در صورت بروز مشکل، آن را در نامه‌ای با عنوانی که با عبارت «CT04» شروع می‌شود به آدرس [gholamirudi@nit.ac.ir](mailto:gholamirudi@nit.ac.ir) گزارش دهید.