



دانشگاه صنعتی شریف  
دانشکده‌ی مهندسی کامپیوتر

# برنامه‌سازی پیشرفته

نیم‌سال دوم ۹۷-۹۸

تمرین یک

مفاهیم مقدماتی جاوا

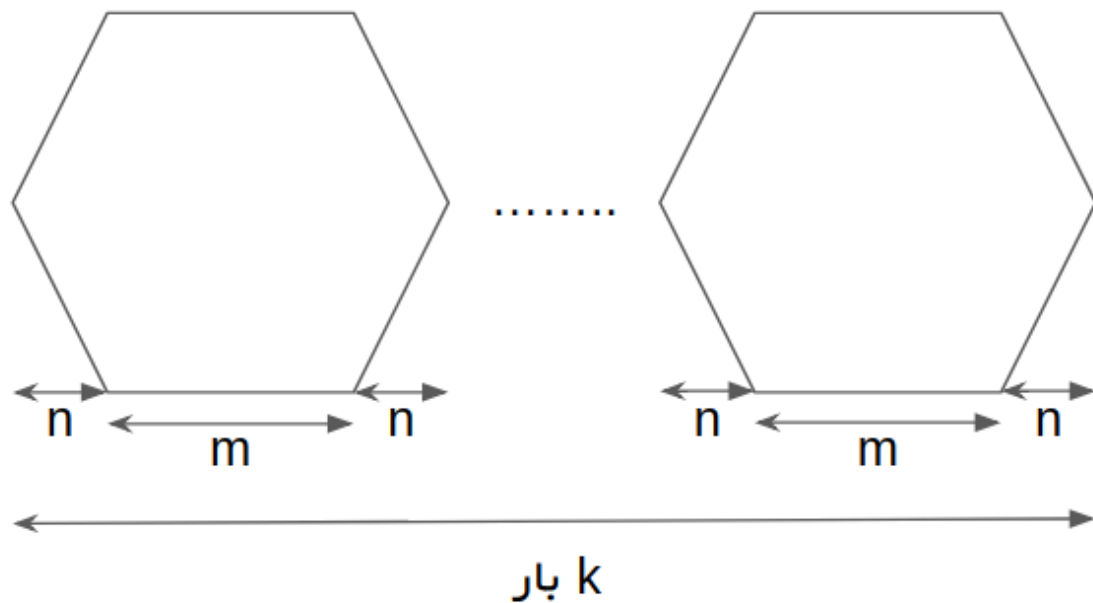
مهلت ارسال: دوشنبه ۲۰ اسفند

به موارد زیر توجه کنید:

- پاسخ تمرین را در سامانه‌ی کوئرا بارگذاری نمایید.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام‌شده است. توصیه می‌شود نوشتن تمرین را به روزهای نهایی موکول نکنید.
- مهلت ارسال پاسخ تمرین تحت هیچ شرایطی تمدید نخواهد شد.
- هم‌کاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ‌های ارسالی هر کس حتماً باید توسط خود او نوشته شده باشد. ارسال پاسخ تمرین‌ها ۴ روز فرصت ارسال با تاخیر و با جریمه‌ی کسر روزانه ۲۰٪ از نمره‌ی سوال در نظر گرفته شده است که برای محاسبه‌ی نمره‌ی سوال در نظر گرفته شده است که برای محاسبه‌ی نمره‌ی تمرین در مجموع ۴ روز از تاخیر تمرین‌ها بخشیده می‌شود.
- به ازای هر روز ارسال زودتر پاسخ هر سوال از تمرین، ۵٪ نمره‌ی مثبت تا سقف ۴ روز (حداکثر ۲۰٪) تعلق خواهد گرفت. (فقط در صورت کسب نمره‌ی کامل سوال که شامل نمره‌ی امتیازی نیز می‌شود)
- مبنای درس، اعتماد به پاسخ‌های ارسالی توسط شماست و در نتیجه در صورت مشاهده‌ی هرگونه مشابهت غیرممتعارف بین کدها، برای هر دو طرف تقلب‌دهنده و تقلب‌گیرنده در مرتبه‌ی اول نمره‌ی ۱۰۰- برای آن تمرین و در صورت تکرار، حذف درس صورت خواهد گرفت.

۱ سوال  $(\lim_{x \rightarrow 0} \sin x / x)$  -ام (۲۰ امتیاز)

در این سوال شما باید ۶-ضلعی مانند شکل با کاراکترهای \* چاپ کنید.



ورودی

ورودی تنها شامل یک خط است که در آن سه عدد طبیعی  $n$  و  $m$  و  $k$  با فاصله از هم آمده است.

$$1 \leq n, m, k \leq 50$$

ورودی	خروجی
1 1 1	* *** *
1 2 1	** ***** **
2 3 2	***      *** *****   ***** ***** *****   ***** ***      ***
1 2 3	**   **   ** ***** **   **   **

## ۲ اشتراک (۲۰ امتیاز)

در این سوال باید اشتراک تعدادی مجموعه را حساب کنید.

### ورودی

در خط اول عدد  $n$  به شما داده می شود که نشان دهنده ی تعداد مجموعه هاست. در خط بعدی  $k_1$  تا  $k_n$  می آیند که  $k_i$  نشان دهنده ی تعداد اعداد در مجموعه ی  $i$  ام می باشد. سپس  $n$  خط در ورودی می آیند که خط  $i$  ام آن ها  $k_i$  عدد دارد که اعضای آن مجموعه را نشان می دهد. تضمین می شود که مجموع تعداد اعضای مجموعه ها از  $10^6$  بیشتر نیست یعنی

$$k_1 + k_2 + \dots + k_n \leq 10^6$$

$$1 \leq a_{ij} \leq 10^9$$

منظور از  $a_{ij}$  عنصر  $j$  ام مجموعه ی  $i$  ام است. هم چنین تضمین می شود اعداد درون یک مجموعه همگی متمایز هستند.

### خروجی

خروجی شما باید یک خط باشد که شامل اشتراک تمامی مجموعه هاست. اعداد خروجی باید با فاصله از هم و به صورت مرتب شده از کوچک به بزرگ باشند.

### مثال

ورودی	خروجی
3 4 5 6 20 10 8 7 7 9 10 20 35 9 8 35 10 7 3	7 10
2 2 3 1 2 3 4 5	

توضیح مثال ۲: اشتراک مجموعه ها تهی است بنابراین خروجی هم شامل هیچ عددی نیست.

### ۳ شایعه (۲۰ امتیاز)

در شهر خیلی خیلی دور  $n$  نفر زندگی می کنند که برخی از آن ها با همدیگر رابطه ی دوستی دارند. خوشبختانه تمام این رابطه ها دو طرفه می باشند.

هدف شما این است که یک شایعه را بین تمام افراد این شهر پخش کنید. برای این کار شما شایعه را به بعضی از افراد شهر می گوئید و آن ها شایعه را بین دوستانشان پخش می کنند و دوستانشان هم این شایعه را برای دوستان خود پخش کرده و شایعه تا جای ممکن پخش می شود.

اما پخش کردن شایعه هزینه دارد و وقتی شما می خواهید به نفر  $i$  ام شایعه را بگوئید باید هزینه  $c_i$  به او بپردازید اما پخش شدن شایعه مجانی است و هر کس که شایعه را شنیده باشد بدون هزینه آن را بین دوستانش پخش می کند. حال شما می خواهید بدانید که کمترین هزینه ی لازم برای این که شایعه بین تمام افراد شهر پخش شود چقدر است.

#### منبع

برای الگوریتم های جستجوی گراف می توانید به لینک های زیر مراجعه کنید.

Graph traversal  
DFS

#### ورودی

در خط اول دو عدد  $n$  و  $m$  می آید که به ترتیب تعداد افراد شهر و تعداد روابط دوستی را نشان می دهد.

$$1 \leq n \leq 10^5, 0 \leq m \leq 10^5$$

در خط بعدی  $n$  عدد  $c_i$  می آید که هزینه ی گفتن شایعه به نفر  $i$  ام را نشان می دهد.

$$0 \leq c_i \leq 10^9$$

سپس در  $m$  خط بعدی در هر کدام دو عدد متمایز  $i$  و  $j$  بین ۱ تا  $n$  میاید که دوستی بین نفر  $i$  ام و  $j$  ام را نشان می دهد.

#### خروجی

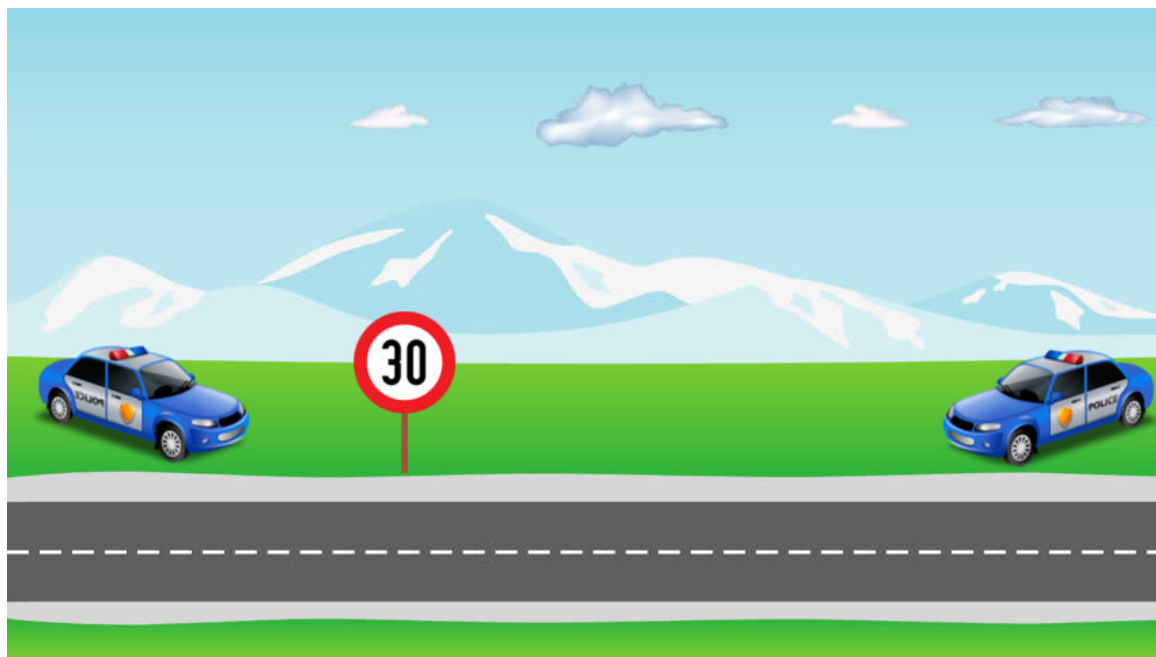
در تنها خط خروجی یک عدد را چاپ کنید (کمترین هزینه ی لازم برای شما که با آن بتوانید شایعه را بین تمام آدم های شهر پخش کنید).

مثال

خروجی	ورودی
10	5 2 2 5 3 4 8 1 4 4 5
55	10 0 1 2 3 4 5 6 7 8 9 10
15	10 5 1 6 2 7 3 8 4 9 5 10 1 2 3 4 5 6 7 8 9 10

## ۴ پلیس‌های بی‌کار (۱۰ + ۲۰ امتیاز)

در جاده ای ۲ پلیس در فاصله ۵۰۰ متری نسبت به هم قرار دارند. در این جاده ماشین‌ها با سرعت زیاد در حال تردد هستند و وظیفه پلیس‌ها ضبط اطلاعات ماشین‌هایی است که با سرعت غیر مجاز تردد می‌کنند. منتها از آنجا که ماشین‌ها سرعتشان سرسام‌آور است؛ پلیس‌ها ممکن است اطلاعات ماشین‌ها را اشتباه دیده و در نتیجه اشتباه یادداشت کنند. پس اطلاعاتی که پلیس‌ها بدست آورده‌اند به احتمالی درست است.



پلیس‌ها اطلاعات خود را به فرمت خیلی مخوفی یادداشت می‌کنند که ما برایتان در چند سطر آن را توضیح می‌دهیم:

- اطلاعات ماشین‌ها با / تمایز داده می‌شوند.
  - اطلاعات یک ماشین به صورت `VEHICLE_NAME.xxx.x.xx.xx` یادداشت می‌شود به این صورت که ابتدا اسم آن ماشین سپس در ۶ حرف پلاک آن ماشین (۳ رقم، یک حرف انگلیسی بزرگ، ۲ رقم) و در انتها یک عدد دورقمی برای نشان دادن احتمال درست بودن اطلاعات مربوط به ماشین فعلی می‌آید.
- به عنوان مثال `gari.123.A.45.07` نشان‌دهنده‌ی ماشینی با نام `gari` با پلاک `123.A.45` و پلیسی که این اطلاعات را وارد کرده است به احتمال ۷ درصد مطمئن بوده است. (چه پلیس بی‌دقتی!) در این جاده از آنجایی که رانندگان باهوش هستند (!! در صورتی که یکی از پلیس‌ها اطلاعات آن‌ها را یادداشت کند متوجه می‌شوند. بنابراین اگر پلیس ۱ ماشینی را جریمه کرده باشد راننده آن ماشین متوجه شده و در ادامه جاده محتاطانه می‌راند تا توسط پلیس ۲ جریمه نشود. پس از گذشت یک روز اطلاعات جمع شده از ۲ پلیس بررسی می‌شود.

### ورودی

در ورودی ۲ خط داده می‌شود که خط ۱ اطلاعات پلیس ۱ و خط ۲ اطلاعات پلیس ۲ است.

### خروجی

در خروجی اطلاعات ماشین‌ها به ترتیب احتمال درستی و در صورت برابری احتمال‌ها به ترتیب اسم نمایش داده می‌شوند. همچنین هر ورودی غیر مجاز باید پیام `"Invalid Input"` را نمایش دهد. توجه داشته باشید در صورتی که اطلاعات یک ماشین دو بار یادداشت شود باید پیام `"Invalid Input"` نمایش داده شود و در صورتی که نام و شماره پلاک دو ماشین یکی باشد دو ماشین برای ما یکسان فرض می‌شود. تا ۱۰ نمره امتیازی برای استفاده از `regex` در نظر گرفته می‌شود.

ورودی	خروجی
pride132.111.A.44.06/peykan.981.B.13.70 lexus.009.C.09.99/honda.999.Z.99.70	lexus->009.C.09 ,probability: 99 honda->999.Z.99 ,probability: 70 peykan->981.B.13 ,probability: 70 pride132->111.A.44 ,probability: 6
geep.123.A.12.56 geep.123.A.12.80	Invalid Input



## ۵ رمزگشایی ساده (۵ + ۲۰ امتیاز)

در این مسئله به شما تعدادی کلید واژه داده می‌شود و از شما خواسته می‌شود که متن‌های رمزگذاری شده را رمزگشایی کنید. این کلیدواژه‌ها نوعی عملگر هستند و باعث تغییر رشته تا رسیدن به رشته مطلوب می‌شوند. این عملگرها عبارت‌اند از:

- عملگر پایان بازی: عملگری است با کلید واژه‌ای ۴ حرفی که در صورت مشاهده‌ی آن در رشته، رشته به همان صورت فعلی چاپ می‌شود و بازی پایان می‌پذیرد. دقت شود که این کلیدواژه در صورتی که تعدادی از حروف آن در یک رشته و تعدادی در رشته‌های بعدی یافت شوند نیز پایان می‌پذیرد. (مرحله‌ی نهایی رشته‌های قبلی و هر مرحله‌ی رشته‌ی فعلی) این عملگر در ابتدا وجود ندارد و برای اولین بار توسط عملگر *CFWord* تعیین می‌شود.
  - عملگر *CFWord* (تغییر کلیدواژه پایان بازی): این عملگر رشته‌ای را تعیین می‌کند که در صورت مشاهده شدن در رشته‌ی اصلی، چهار حرف بعد از آن به عنوان رشته‌ی پایان‌دهنده به بازی تعیین می‌شوند و سپس به همراه این چهار حرف از رشته حذف می‌شود. (تضمین می‌شود که بعد از آن چهار حرف وجود دارد)
  - عملگر *SWord* (توقف تغییرات): این عملگر کلیدواژه و تعداد تکرار آن را تعیین می‌کند که در صورت مشاهده‌ی این کلیدواژه در رشته به تعداد داده شده، از ساده شدن بیشتر رشته جلوگیری می‌کند (وجود کلید واژه‌ی عملگر پایان بازی چک می‌شود) و پس از حذف همین تعداد از کلیدواژه، رشته چاپ می‌شود.
  - عملگر *Cpy* (کپی): در صورت مشاهده شدن این کلید واژه در رشته، بجای حروف خود کلیدواژه، حروف بعد از آن را نمایش می‌دهد، و در صورتی که پس از کلید واژه *Cpy* عدد باشد (اعداد یک رقمی ۰ تا ۹)، به تعداد عدد حروف جایگزین کلیدواژه و عدد می‌شود. (مثال: برای کلید واژه *df* نتیجه‌ی *abdfcd* می‌شود *abcdcd* و نتیجه‌ی *thrd1e* می‌شود *three*)
  - عملگر *Mul* (ضرب): در صورت مشاهده اگر در دو سمت آن عدد وجود داشت (اعداد با هر تعداد رقم)، حاصل ضرب آن دو عدد را به جای آن دو عدد قرار می‌دهد (مثال: برای کلیدواژه *abc* نتیجه‌ی *s11abc2ef* برابر با *s22ef* می‌شود) و در صورت نبودن عدد اتفاقی نمی‌افتد.
  - عملگر *Add* (جمع): مانند عملگر ضرب با این تفاوت که دو عدد را جمع می‌کند.
  - عملگر تغییر اعداد از صورت انگلیسی به عددی: در صورتی که اعداد به صورت حروف انگلیسی نوشته شده باشند، به صورت عددی آن‌ها تبدیل می‌شوند (فقط برای اعداد تک رقمی صفر تا نه) و این عملگر همیشه از کمترین اولویت برخوردار است. (مثال: نتیجه‌ی *rfeightwo* تبدیل می‌شود به *rf8wo*)
- به صورت کلی کلیدواژه‌ها به بزرگ یا کوچک بودن حروف انگلیسی وابسته نیستند، همچنین عملگرها به ترتیب اولویت اعمال می‌شوند و پس از همه‌ی تغییرات یک عملگر، وجود عملگرهایی با اولویت بیشتر دوباره بررسی می‌شود. اولویت اعمال برای کلیدواژه‌هایی با اولویت برابر با کلیدواژه‌هایی است که به ابتدای رشته نزدیک‌ترند. (مانند مثال قبل)
- تا ۵ نمره امتیازی برای استفاده از تابع برای هر عملگر در نظر گرفته می‌شود**

### ورودی

در خطوط فرد کلید واژه‌ها داده می‌شوند که ترتیب داده شدن آن‌ها، ترتیب اولویت آن‌ها را مشخص می‌کند و مانند رو به رو داده می‌شوند (میان آن‌ها همیشه "،" و فاصله وجود دارد و ترتیب عملگرها می‌تواند عوض شود):  
*CFWord*: (کلید واژه آن)، *Cpy*: (کلید واژه آن)، *Sword*: (تعداد تکرار) (کلید واژه آن)، *Mul*: (کلید واژه آن)، *Add*: (کلید واژه آن)  
 در خطوط زوج نیز رشته‌ای که باید رمزگشایی شود داده می‌شود. این رشته و کلید واژه‌ها فقط از حروف بزرگ و کوچک انگلیسی تشکیل شده‌اند (بدون فاصله). همچنین تضمین می‌شود که تعداد خطوط از ۲۰، طول کلید واژه‌ها از ۱۰ و طول رشته از ۱۰۰ بیشتر نیست. (حداقل طول آن‌ها نیز یک است).

### خروجی

در خط  $n$  ام خروجی رشته‌ی ساده شده و به صورت حروف کوچک انگلیسی خط  $2n$  ام ورودی چاپ می‌شود.

## مثال

خروجی	ورودی
3 isnot even	Cpy: sv7, Mul: mul, Add: eene, SWord: bye 2, CFWord: evee threSsvseven4Vsevs7eNeoNeMulfourh Mul: hel, Add: this, CFWord: butyo, SWord: ali 3, Cpy: sn AliIsAliButYouAreNotAli Mul: hehe, Add: haro, Cpy: change4, CFWord: end, SWord: dotch 1 endotchange42even

**توضیح مثال:** رشته‌ی اول به ترتیب به *thressvseven4vseveneeneone mul fourh* (عملگر کپی)، *thressv74v7eene1 mul 4h* (عملگر تبدیل نوشتاری به عددی اعداد)، *threeeveev7eene1 mul 4h* (عملگر ضرب)، *threeeveev11h* (عملگر جمع)، به *three* (عملگر تغییر کلمه پایان برنامه) و در آخر به ۳ (عملگر تبدیل نوشتاری به عددی اعداد) تبدیل می‌شود. رشته‌ی دوم هم پس از تبدیل به حروف کوچک به *aliisalinali* (عملگر تغییر کلمه پایان برنامه) و سپس به *isnot* (عملگر توقف تغییرات) تبدیل می‌شود و بیش از آن ساده نمی‌شود. رشته‌ی سوم هم پس از تبدیل به *endoteveeven* (عملگر کپی) به *even* (عملگر تغییر کلمه پایان برنامه) تبدیل می‌شود و سپس برنامه پایان می‌پذیرد. (کلمه پایان بازی در آخرین تغییر به *otev* تغییر کرده بود که حروف *ot* در رشته‌ی قبلی و حروف *ev* نیز در ابتدای رشته‌ی فعلی وجود دارند).