## سوال ۴. *Con Respecto*

سپهر که به تازگی خیلی در Blockchain و Smart Contract خفن شده است و به استخدام فردی به نام هانی در همین زمینه درآمده و هفته ای دست کم دو پروژه بلاک چین را در برنامه هفتگی خود دارد. هانی که فردی جاه طلب است می‌خواهد یک miner بخرد و کد آن را دست‌کاری کند تا بتواند کمی هم از شبکه بلاک‌چین پولشویی کند. از آنجا که خودش عرضه این کار ها را ندارد آن‌را به سپهر می‌سپارد.

بیش ازین حرف نزنیم و بگیم، تو باید چه چیز هایی بدونی، و چه کاری از تو انتظار می‌ره:

از بدو خلقت تا یگانه لحظه اکنون که این متنو می‌خونی m بلاک در شبکه ساخته شده که به صورت زنجیر وار به هم متصل شده‌اند (blockchain) . درون هر بلاک n تراکنش (transaction) وجود دارد (ممکن است تعداد تراکنش های یک بلاک با بلاکی دیگر فرق کند!) که هر تراکنش متناظر با یک رشته است. هر یک از بلاک ها نیز رشته ای معرفی کننده و بسیار مهم به نام hash دارد که این رشته تابعی از تمام تراکنش های موجود در بلاک و همچنین hash بلاک قبلی است(البته ما در این سوال فرضیاتی برای ساده سازی مسیله انجام دادیم که در واقعیت به این شکل نیست:). یعنی برای تولید hash بلاک i ام به تمام تراکنش های این بلاک و همچنین hash بلاک i-۱ ام نیاز داریم.(قطعا باید تا الان نگران تولید hash مربوط به بلاک شماره ۱ خلقت شده باشی، نگران نباش اون hash رو ما به شما می‌دیم.)

**اما چه ترکیب خاصی از تراکنش ها و hash قبلی، hash جدید را می‌سازد؟**

ابتدا تابع $value(s : string, P : int)$ را به صورت زیر تعریف می‌کنیم:

$$value(s, P) = ((\sum_{k=0}^{len(s)-1} s[k].P^k) mod 94) + 33$$

ورودی های تابع *value*:

- اول s که یک رشته است و s[k] که کد ASCII نظیر به کاراکتر kام از آن رشته است که در محاسبات استفاده می‌شود.
- دوم P یک عدد اول منسوب به بلاک است؛ به این معنا که برای هر بلاک می‌تواند با

بلاک دیگر فرق کند.

خروجی تابع *value*:

- خروجی عملیات ریاضی فوق منطقا یک عدد است که تابع value آنرا به کارکتر نظیر به کد اسکی‌ای برابر همان عدد تبدیل می‌کند و return می‌کند. تضمین می‌شود این عدد قطعا کارکتر منطقی‌ای می‌سازد و در بازه معقولی از کد های ASCII قرار می‌گیرد(به این فکر کن که چجوری تضمین می‌شه!)

**تابع value دو جا استفاده می‌شود:**

اول: برای بدست آوردن Primary String مربوط به بلاک. اینکه Primary String کجا استفاده می‌شه رو یکم دیگه بهت می‌گیم ولی اینکه چجوری این رشته رو برای یه block بسازی:

```
block.primary_string[i] = value(block.transactions[i], block.P)
```

همانطور که معلومه تابع value، کاراکتر iم از رشته Primary String یک بلاک را با عمل کردن روی تراکنش iم همان بلاک و با استفاده از عدد اول (P) همان بلاک را می‌سازد.

دوم: حال رسیدیم به **اصل کار** (ویلسون)، بدست آوردن hash بلاک. hash همون Primary String مربوط به بلاک است. فقط یه کارکترشو باید عوض کنی (جایی که hash بلاک قبلی استفاده می‌شه). سوالایی که باید واست ایجاد شده باشه اینه: کدوم کارکترو عوض کنم؟ جاش چی بذارم؟

ابتدا دومیو جواب می‌دیم. جاش alter رو بذار:

$$alter = value(previous\_hash, block.P)$$

که مسلما `previous_hash` همون هش بلاک قبلی‌است.

حالا سوال اولت، کارکتر alter رو بذار جا اندیس زیر:

$$alter \bmod block.n$$

که block.n تعداد تراکنش های همین بلاکیه ک داری زور می‌زنی hash اون رو حساب کنی.

## اجبار سپهر

سپهر که گنگش بالاتر ازیناست که بخواهد کد کثیف بزند ما را مجبور کرد بهت شی گرایی یاد بدیم. ولی ما که می‌دونیم اگ حرفی از شی گرایی وسط بکشیم باید قید تی‌ای بودنو بزنیم صداشو در نمیاریم ولی بجاش ازت می‌خواهیم عناصر زیر رو در ابتدای کدت قرار بدی:

```
    struct block {
    /* block variables */
};

typedef struct block Block;
Block* new_block(int, int);
```

پیاده سازی محتوای ساختاری block که متناسب سوال باشد بر عهده شماست ولی تابع `new_block` چیست؟ پروتوتایپ این تابع به شما داده شده است و پیاده سازی آن تابع هم به عهده خود شماست. شما باید به گونه ای آنرا پیاده سازی کنید که هر گاه نیاز به یک متغیر از نوع block struct داشتید آن تابع را صدا بزنید و برایتان یک متغیر ازین نوع بسازد و پوینتری از آنرا برگرداند. **بسیاار مهم است که بدانید** هر جای کد، غیر از درون تابع `new_block` سعی کنید به هر شکلی یک متغیر از نوع struct block بسازید، **سپهر در نهایت که کد ها را چک می‌کند ۵۰ درصد نمره دریافتی‌تان ازین سوال رو برای شما در نظر نمی‌گیرد حتی اگر کدتان از کوئرا نمره کامل گرفته باشد** چرا که انتظار می‌رود همانطور که گفته شد برای ساختن یک بلاک جدید در هر جای دیگر کد فقط از خروجی تابع `new_block` استفاده کنید. ترم بعد می‌بینید چیزی که زدید شدیدا شبیه مفهوم Constructor ها در شی گرایی‌ست!

## ورودی

سپهر سرش خیلی شلوغه چند تا پروژه دیگه‌ست. همینطور که می‌بینی به ما گفته داک این پروژه کوچیکو براتون بنویسیم و حتی خودش وقت نکرده بیاد بهت بلاک‌چینو توضیح بده. ما در ابتدا در یک خط به شما m را می‌دهیم که تعداد بلاک هاییه ک مونده رو دستمون. تضمین می‌کنیم m عددی مثبت است.

در خط بعد رو یک خط کامل بدون space بهت hash بلاک اول این دسته بلاک جا مونده رو می‌دیم. پس از دادن hash بلاک اول $m-1$ بار اطلاعات بلاک های ۲ تا m را

به شما می‌دهیم. اطلاعات هر بلاک به صورت زیر است:

در یک خط به ترتیب ابتدا n و سپس P بلاک داده می‌شود. یه تضمین دیگه هم می‌کنیم که P عدد اول است. در نهایت در n خط بعدی n رشته بدون space به عنوان رشته های تراکنش های ۱ تا n مربوط به این بلاک به شما داده می‌شود.

## خروجی

لطف کن و hash مربوط به بلاک آخر ( m م ) رو تو یه خط چاپ کن ما هم بریم به بقیه کارامون برسیم.

## مثال

**ورودی نمونه ۱**

```
2
abc
2 7
defghijklm
nopqrstuv
```

(از خط اول می‌فهمیم) ۲ بلاک داریم که (از خط دومش می‌فهمیم) hash بلاک اول که قول دادیم بهتون بدیم abc است. بلاک دوم مشخصاتش به شکلیه ک اومده. ۲ تا تراکنش داره و عدد اولش هم P=۷ عه. تو دو خط بعد اون هم رشته مربوط به تراکنش های اول و دومِ این بلاک اومده.

**خروجی نمونه ۱**

```
ny
```

بلاک آخر، بلاک دومه که اگ Primary String شو بر اساس عدد اولش و تراکنش هاش حساب کنی رشته بدست اومده nw خواهد بود. نتیجه حاصل از پاس دادن hash قبلی (abc) به تابع value هم حرف y با کد اسکی ۱۲۱ خواهد بود. $121\%2 = 1$ پس، از رشته پرایمری بدست آمده برای بلاک آخر، اندیس ۱ را با y جا گذاری می‌کنیم که میشه همین hash ی که بالا نوشته شده است.