# کفترها (Parrots)

یانی (Yanee) یک کفترباز است. از زمانی که یانی راجع به سیستم IPoAC (مخفف «انتقال اطلاعات با استفاده از پرندگان») چیزهایی شنیده است مصمم شده تا یک گروه از کبوتران هوشمند را برای انتقال پیام‌هایی به مقصدهای دوردست تربیت کند.

رؤیای یانی این است که بتواند از پرندگانش برای ارسال یک پیام $M$ به یک سرزمین خیلی خیلی دور استفاده کند. پیام $M$ که یانی قصد ارسال آن را دارد، از $N$ عدد صحیح (نه لزوماً متفاوت) در محدوده‌ی صفر تا ۲۵۵ (شامل این دو عدد) تشکیل شده است. یانی $K$ عدد کفتر که به‌صورت ویژه تعلیم داده شده‌اند، برای این منظور دارد. منتهی تمام پرنده‌ها کاملاً شبیه هم هستند و یانی نمی‌تواند بین آن‌ها تمایزی قائل شود. هر کدام از این کفترها هم، تنها یک عدد صحیح بین صفر تا $R$ (شامل این دو عدد) را می‌تواند به‌خاطر سپرده و حمل کند.

در ابتدا یانی یک مدل بسیار ساده را پیاده‌سازی کرد: او برای ارسال پیام، پرنده‌ها را یک به یک از قفس‌شان بیرون آورده و به هر یک از آن‌ها یک عدد از دنباله را به‌ترتیب آموزش داد. سپس کفتران را به پرواز در آورد. بدبختانه، این روش جواب نداد؛ چرا که، پرندگان الزاماً به‌همان ترتیبی که پرواز را شروع کرده بودند، به مقصد نرسیدند. نتیجتاً در این روش، یانی می‌توانست تمامی اعداد را بازیابی کند، منتهی نمی‌توانست ترتیب درست بین این اعداد را بیابد.

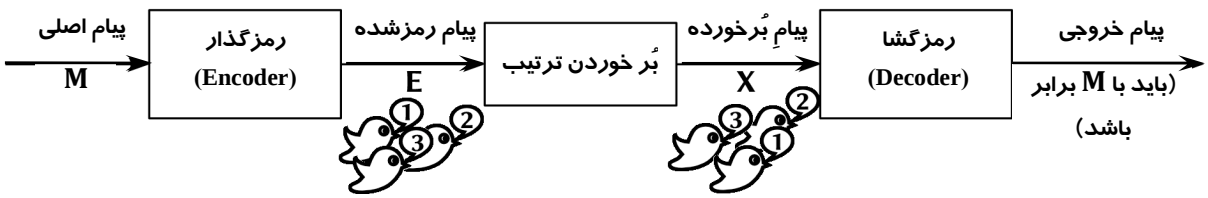
برای تحقّق رؤیای یانی، او به مدل بهتری نیاز دارد و به همین دلیل از شما کمک می‌خواهد. با داشتن پیام $M$، یانی همواره ابتدا کفتران را یکی یکی از قفس درآورده و مشابه روش قبل پس از آموختن یک عدد به هر کدام‌شان، آن‌ها را به پرواز در می‌آورد. یانی از شما تقاضا دارد که برنامه‌ای را بنویسید که دو عمل مجزای زیر را انجام دهد:

- به‌عنوان عمل اوّل، برنامه شما باید یک پیام $M$ را خوانده و آن را به دنباله‌ای از حداکثر $K$ تا عددِ صحیح بین صفر تا $R$ (شامل این دو عدد) تبدیل کند. یانی این اعداد را به پرندگان خواهد آموخت.
- به‌عنوان عمل دوم، برنامه شما باید یک لیست از اعداد بین صفر تا $R$ (شامل این دو عدد) را که اعدادی هستند که کفتران به مقصد رسانده‌اند، دریافت کرده و پس از تحلیل، پیام اصلی $M$ را از آن استخراج کند.

می‌توانید فرض کنید که تمام کفتران همیشه به مقصد می‌رسند و هر کدام از آن‌ها هم عدد اختصاصی‌اش را به‌درستی به‌خاطر می‌سپارد. یانی یک بار دیگر به شما یادآور می‌شود که کفتران ممکن‌ست به هر ترتیبی به مقصد برسند. توجه داشته باشید که یانی $K$ تا کفتر بیشتر ندارد و از همین رو، دنباله‌ای از اعداد صحیح بین صفر تا $R$ که شما تولید می‌کنید، باید حداکثر شامل $K$ تا عدد باشد.

## وظیفه شما

شما باید دو روال مجزّا بنویسید. یکی از آن‌ها به‌عنوان ارسال کننده (رمزگذار یا encoder)، و دیگری به‌عنوان دریافت‌کننده (رمزگشا یا decoder) استفاده می‌شود. کلِّ پروسه در شکل زیر نمایش داده شده است.



دو روالی که شما می‌بایست بنویسید به شرح زیر هستند.

- روال **encode(N, M)** پارامترهای زیر را می‌گیرد:
  - **N** : طول پیام اصلی
  - **M** : یک آرایه‌ی یک‌بُعدی از $N$ عدد صحیح که تشکیل‌دهنده‌ی متن پیام هستند. می‌توانید فرض کنید که برای هر $0 \le i < N$ الزاماً $0 \le M[i] \le 255$ می‌باشد.

این روال باید پیام $M$ را به دنباله‌ای از اعداد صحیح بین صفر تا $R$ (شامل این دو عدد) که قرار است توسّط کفتران ارسال شوند، تبدیل کند. برای گزارش کردنِ (report کردن) این دنباله‌ی تبدیل شده، روال encode می‌بایست روال **send(a)** را برای هر عدد **a** ای که می‌خواهید این a به دقیقاً یک کفتر داده شود، یک بار صدا کند.

- روال **decode(N, L, X)** پارامترهای زیر را می‌گیرد:
  - **N** : طول پیام اصلی
  - **L** : طول پیام دریافتی (تعداد کفترهایی که ارسال شده‌اند)
  - **X** : یک آرایه‌ی یک‌بُعدی از $L$ عدد صحیح، که اعداد دریافت‌شده را نشان می‌دهند. اعداد $X[i]$ برای $i$های بین $0 \le i < L$، دقیقاً اعدادی هستند که روال encode شما آن‌ها را تولید کرده، منتهی ترتیب‌شان احتمالاً تغییر کرده است.

این روال می بایست پیام اصلی را بازیابی کند. برای گزارش کردنِ این پیامِ بازیابی شده، روال decode می‌بایست روال **output(b)** را برای اعداد صحیح **b** در پیامِ بازیابی شده به‌ترتیب فراخوانی کند.

توجه کنید که $R$ و $K$ به‌عنوان پارامتر ورودی داده نمی‌شوند. توضیحات زیرمسئله‌ها را در ادامه ببینید.

برای این که شما یک زیرمسئله را کامل حل کنید، شروط زیر می‌بایست در مورد روال‌های شما صادق باشد:

- تمامی اعدادی که توسط روال encode شما به سیستم ارسال می‌شوند، می‌بایست در محدوده‌ی تعیین شده‌ی زیرمساله باشند.
- تعداد دفعاتی که روال encode روال send را فرا می‌خواند نباید از حدّ مشخص شده در زیر مسئله ($K$) بیش‌تر بشود. توجه کنید که $K$ به طول پیام وابسته است.
- روال decode می‌بایست دقیقاً و به‌طور صحیح پیام اصلی $M$ را بازیابی کرده و روال output(b) را دقیقاً $N$ بار برای b مساوی با $M[0]$، $M[1]$، ... و $M[N-1]$ به‌ترتیب صدا بزند.

در آخرین زیرمسئله، نمره‌ی شما وابسته به نسبت طول پیام رمزگذاری شده به طول پیام اصلی است.

## مثال نمونه

حالتی را در نظر بگیرید که $N = 3$ بوده و پیام $M$ به‌صورت مقابل باشد.

$$M = \begin{matrix} 10 \\ 30 \\ 20 \end{matrix}$$

فرض کنید در این حالت، روال encode$(N, M)$ با استراتژی ناشناخته‌ای این پیام را به دنباله‌ی $\langle 7,3,2,70,15,20,3 \rangle$ رمزگذاری (encode) کند. برای ارسال این دنباله، برنامه می‌بایست روال send را به‌ترتیب زیر فراخوانی کند:

```
send(7)
send(3)
send(2)
send(70)
send(15)
send(20)
send(3)
```

حال فرض کنید که زمانی‌که تمامی کفتران به مقصد خود رسیدند، ما لیست اعداد را به‌صورت $\langle 3,20,70,15,2,3,7 \rangle$ دریافت می‌کنیم. اکنون روال decode با پارامترهای $N = ۳$ و $L = ۷$ و $X$ به‌صورت روبه‌رو فراخوانی می‌شود:

$$X = \begin{matrix} 3 \\ 20 \\ 70 \\ 15 \\ 2 \\ 3 \\ 7 \end{matrix}$$

روال decode می‌بایست با داشتن این اطلاعات پیام اصلی $\langle 10,30,20 \rangle$ را بسازد. برای گزارش می‌بایست روال output به ترتیب زیر صدا زده شود.

```
output(10)
output(30)
output(20)
```

## زیرمسئله‌ها

### زیرمسئله‌ی شماره یک (۱۷ امتیاز)

- $N = ۸$ است و هر یک از اعداد آرایه‌ی $M$ یا صفر و یا یک هستند.
- هر عدد رمزگذاری شده باید در محدوده‌ی **صفر الی** $R = ۶۵۵۳۵$ (شامل این دو عدد) باشد.
- تعداد دفعاتی که شما می‌توانید روال send را فراخوانی کنید حداکثر $K = ۱۰ \times N$ است.

### زیرمسئله‌ی شماره دو (۱۷ امتیاز)

- $۱ \le N \le ۱۶$ است.
- هر عدد رمزگذاری شده باید در محدوده‌ی **صفر الی** $R = ۶۵۵۳۵$ (شامل این دو عدد) باشد.
- تعداد دفعاتی که شما می‌توانید روال send را فراخوانی کنید حداکثر $K = ۱۰ \times N$ است.

### زیرمسئله‌ی شماره سه (۱۸ امتیاز)

- $۱ \le N \le ۱۶$ است.
- هر عدد رمزگذاری شده باید در محدوده‌ی **صفر الی** $R = ۲۵۵$ (شامل این دو عدد) باشد.
- تعداد دفعاتی که شما می‌توانید روال send را فراخوانی کنید حداکثر $K = ۱۰ \times N$ است.

### زیرمسئله‌ی شماره چهار (۲۹ امتیاز)

- $۱ \le N \le ۳۲$ است.

- هر عدد رمزگذاری شده باید در محدوده‌ی **صفر الی** $\boldsymbol{R} = ۲۵۵$ (شامل این دو عدد) باشد.
- تعداد دفعاتی که شما می‌توانید روال send را فراخوانی کنید حداکثر $\boldsymbol{K} = ۱۰ \times \boldsymbol{N}$ است.

### زیرمسئله‌ی شماره پنج (حداکثر ۱۹ امتیاز)

- $۱۶ \leq \boldsymbol{N} \leq ۶۴$ است.
- هر عدد رمزگذاری شده باید در محدوده‌ی **صفر الی** $\boldsymbol{R} = ۲۵۵$ (شامل این دو عدد) باشد.
- تعداد دفعاتی که شما می‌توانید روال send را فراخوانی کنید حداکثر $\boldsymbol{K} = ۱۵ \times \boldsymbol{N}$ است.
- **نکته مهم:** امتیاز این زیرمسئله وابسته به نسبت بین طول پیام رمزگذاری شده و طول پیام اصلی است.

برای یک تست $\boldsymbol{t}$ در این زیرمسئله، $\boldsymbol{P_t = L_t/N_t}$ را نسبت بین طول پیام رمزگذاری $(L_t)$ و طول پیام اصلی $(N_t)$ می‌گیریم. فرض کنید $\boldsymbol{P}$ برابر با بیشینه‌ی تمام $P_t$ ها باشد. در این صورت امتیاز شما در این زیرمسئله با قوانین زیر محاسبه می‌شود:

- اگر $P \leq ۵$ باشد، شما امتیاز کامل این زیرمسئله یعنی ۱۹ امتیاز را می‌گیرید.
- اگر $۵ < P \leq ۶$ باشد، شما ۱۸ امتیاز می‌گیرید.
- اگر $۶ < P \leq ۷$ باشد، شما ۱۷ امتیاز می‌گیرید.
- اگر $۷ < P \leq ۱۵$ باشد، امتیاز شما برابر با $۱ + ۲ \times (۱۵ - P)$ (رُند شده به پایین) خواهد بود.
- اگر $P > ۱۵$ باشد یا حداقل یکی از خروجی‌های شما نادرست باشد، شما صفر می‌گیرید.

**نکته مهم:** هر راه‌حل معتبری برای زیرمسئله‌های یکم الی چهارم، تمامی زیرمسئله‌های قبلی‌اش را هم حل می‌کند. اگرچه با عنایت به محدوده‌ی بزرگتر برای $K$ در زیرمسئله‌ی پنجم، یک راه‌حل معتبر برای زیرمسئله‌ی پنجم ممکن‌ست زیرمسئله‌های یکم تا چهارم را حل نکند، ولی راه حلی هست که همه زیرمساله‌ها (یکم تا پنجم) را با هم حل می‌کند.

## جزئیات پیاده‌سازی

### محدودیت‌ها

- **محیط ارزشیابی:** در محیط ارزشیابی، ارسال‌های شما در دو برنامه‌ی مجزای **e** و **d** کامپایل شده و به‌صورت مجزا اجرا می‌شوند. هر دو روال رمزگذار و رمزگشا (encoder و decoder) به هر دو برنامه‌ی اجرایی لینک می‌شوند، منتهی e تنها روال encode و d تنها روال decode را صدا می‌زند.
- **محدودیت زمانی CPU:** برنامه‌ی **e** به‌تعداد ۵۰ دفعه روال encode را فرا می‌خواند و این کار باید در ۲ ثانیه انجام شود. برنامه‌ی **d** نیز ۵۰ دفعه روال decode را فرا می‌خواند و این کار نیز باید در ۲ ثانیه انجام شود.
- **محدودیت حافظه:** ۲۵۶ مگابایت
- **توجه:** هیچ محدودیت صریحی روی اندازه‌ی حافظه‌ی پشته (stack) استفاده شده وجود ندارد. میزان حافظه‌ی استفاده شده در پشته به‌عنوان بخشی از حافظه‌ی مصرفی محاسبه می‌شود.

### واسط API

- فولدرِ پیاده‌سازی: /parrots

- فایل‌هایی که توسط شرکت‌کننده می‌بایست پیاده‌سازی شوند:
  - encoder.c یا encoder.cpp یا encoder.pas
  - decoder.c یا decoder.cpp یا decoder.pas

**توجه برای برنامه‌نویسان به‌زبان C/C++**: هر دو فایل encode.c[pp] و decode.c[pp] شما در مصحح نمونه و در مصحح اصلی با هم به مصحح لینک می‌شوند. بنابراین، شما می‌بایست تمام متغیرهای عمومی در هر فایل را به‌صورت `static` تعریف کنید تا از تداخل آن‌ها با متغیرهای دیگر فایل‌ها اجتناب شود.

- واسط‌های شرکت‌کننده:
  - encoder.h یا encoder.pas
  - decoder.h یا decoder.pas
- واسط مصحح:
  - encoderlib.h یا encoderlib.pas
  - decoderlib.h یا decoderlib.pas
- مصحّحِ نمونه: grader.c یا grader.cpp یا grader.pas

مصححِ نمونه، ۲ دور برنامه‌ی شما را اجرا می‌کند. در هر دور، ابتدا روال `encode` شما با داده‌های ورودی را اجرا کرده و سپس روی خروجی تولید شده توسط این روال، روال `decode` را اجرا می‌کند. در دور اول، مصحح ترتیب اعداد صحیح در پیام رمزگذاری شده را تغییر نمی‌دهد. منتهی در دور دوم، مصحح جای اعداد در موقعیت‌های زوج با فرد را swap می‌کند. مصحح اصلی جایگشت‌های متفاوت دیگری را روی پیام رمزگذاری شده اِعمال می‌کند. شما می توانید روش بُر زدن داده‌ها توسط مصحح نمونه را در روال `shuffle` (در C/C++) و در روال `Shuffle` (در پاسکال) تعویض کنید.

همچنین مصحح نمونه هم بر روی محدوده و هم طول داده‌های رمزگذاری شده، چک‌ای انجام می‌دهد. به‌طور پیش‌فرض مصحح نمونه بررسی می‌کند که اولاً هر عدد صحیح در داده‌های رمزگذاری شده بین صفر تا ۶۵۵۳۵ (شامل هر دو این اعداد) باشد و طول نیز حداکثر $N \times ۱۰$ باشد. شما می‌توانید این دو میزان را با تغییر ثابت `channel_range` (از ۶۵۵۳۵ به ۲۵۵ برای مثال) و `max_expansion` (از ۱۰ به ۱۵ یا ۷، برای مثال) انجام دهید.

- ورودی مصحّح نمونه: grader.in.1، grader.in.2 و ...
- **توجه:** مصحح نمونه، ورودی را به فرمت زیر می‌خواند:
  - **خط اوّل:** مقدار $N$
  - **خط دوم**: لیستی از $N$ عدد که به‌ترتیب $M[0]$، $M[1]$، ... و $M[N-۱]$ هستند.
- خروجی مورد انتظار برای ورودی مصحح نمونه: grader.expect.1، grader.expect.2 و ...

برای این مسئله، هر کدام از این فایل‌ها دقیقاً باید شامل عبارت «**Correct.**» باشند.