

Papagoid

Yanee on linnuhuviline. Pärast *IP over Avian Carriers* (IPoAC) kohta lugemist on ta kulutanud palju aega, et treenida parve intelligentseid papagoisid sõnumeid pikkade vahemaade taha kandma.

Yanee unistus on kasutada oma linde, et saata sõnum M kaugele maale. Tema sõnum M on N täisarvust (mitte ilmtingimata erinevatest) koosnev järjend, igaüks lõigust 0 kuni 255. Yaneel on K papagoid. Kõik papagoid näevad välja ühesugused – Yanee ei suuda neil vahet teha. Iga linn on võimeline meeles pidama ühte täisarvu lõigust 0 kuni R .

Alguses proovis ta lihtsat skeemi: sõnumi saatmiseks laskis Yanee linnud ühekaupa puurist välja. Enne iga linnu õhkutõusmist õpetas ta sellele ühe arvu sõnumist nende sõnumis esinemise järjekorras. Kahjuks see skeem ei töötanud. Kuigi kõik linnud jõudsid lõpuks sihtkohta, ei pruukinud nad seda teha õhkutõusmise järjekorras. Selle skeemi järgi suutis Yanee taastada kõik saadetud arvud, aga ta ei olnud võimeline neid õigesse järjekorda panema.

Oma unistuse teostamiseks vajab Yanee paremat skeemi ja selleks vajab ta teie abi. Sõnumi M saatmiseks kavatseb ta linnud lasta puurist välja ükskhaaval, nagu varemgi. Teid vajab ta kirjutamaks programmi, mis sooritab kahte operatsiooni:

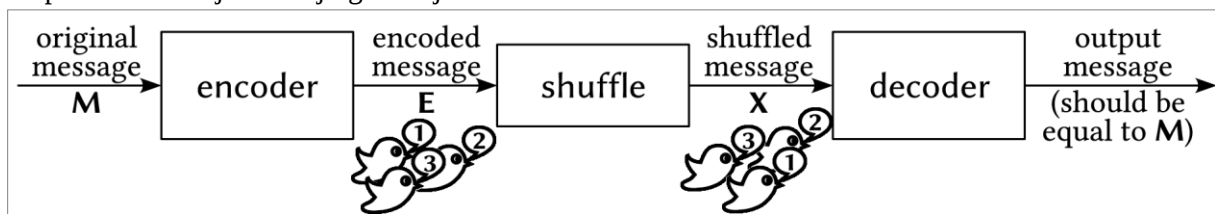
- ✦ Esiteks peab teie programm lugema sõnumi M ja teisendama selle ülimalt K täisarvust $0 \dots R$ koosnevaks järjendiks, mille Yanee lindudele õpetab.
- ✦ Teiseks peab teie programm lugema täisarvude $0 \dots R$ järjendi, mis on saadud lindudelt selles järjekorras, nagu nad sihtpunkti jõudsid.

Võite eeldada, et kõik papagoid jõuavad alati sihtpunkti ja igaüks neist mäletab talle antud arvu. Yanee tuletab teile veelkord meelde, et papagoid võivad sihtpunkti saabuda suvalises järjekorras. Lisaks pange tähele, et Yaneel on ainult K papagoid, nii et teie genereeritud järjend võib koosneda ülimalt K arvust $0 \dots R$.

Ülesanne

Kirjutage kaks meetodit. Ühte neist (encoder) kasutakse saatja, teist (decoder) vastuvõtja poolt.

Terve protsess on kirjeldatud järgneval joonisel.



Kaks meetodit, mille peate kirjutama, on:

- ✦ Meetod **encode**(N, M), millele antakse ette järgmised parameetrid:
 - ✦ N – sõnumi pikkus.
 - ✦ M – sõnumit kirjeldav N täisarvust koosnev ühemõõtmeline massiiv; võite eeldada, et $0 \leq M[i] \leq 255$ iga $0 \leq i < N$ korral.

See protseduur peab kodeerima sõnumi M täisarvude $0 \dots R$ järjendiks, mis papagoidega teele saadetakse. Selle sõnumi esitamiseks peab teie meetod `encode` välja kutsuma meetodi `send(a)` iga täisarvu a korral, mille te soovite ühele lindudest anda.

- ✦ Meetod **decode**(N, L, X), millele antakse ette järgmised parameetrid:



- ⤴ **N** – esialgse sõnumi pikkus.
- ⤴ **L** – vastuvõetud sõnumi pikkus (saadetud lindude arv).
- ⤴ **X** – vastuvõetud järjendit kirjeldav **L** täisarvust koosnev ühemõõtmeline massiiv; arvud **X[i]** üle $0 \leq i < L$ on täpselt need arvud, mille teie encode meetod väljastas, aga nad võivad olla teistsuguses järjekorras.

See meetod peab taastama esialgse teate. Selle esitamiseks peab teie decode meetod õiges järjekorras välja kutsuma protseduuri output(b) iga täisarvu **b** jaoks dekodeeritud sõnumis.

Pange tähele, et **R** ja **K** ei ole parameetritena antud. Nende kohta lugege tagapool alamülesannete kirjeldustest.

Ühe alamülesande õigesti lahendamiseks peavad teie meetodid rahuldama järgmisi tingimusi:

- ⤴ Kõik teie encode meetodi saadetud arvud peavad mahtuma alamülesandes määratud piiridesse.
- ⤴ Teie encode meetod ei tohi meetodit send välja kutsuda rohkem kui alamülesandes määratud **K** korda. Pange tähele, et **K** väärtus sõltub sõnumi pikkusest.
- ⤴ Meetod decode peab õigesti taastama esialgse sõnumi **M** ja kutsuma meetodit output(b) välja täpselt **N** korda, nii et b väärtus on vastavalt **M[0]**, **M[1]**, ..., **M[N-1]**.

Viimases alamülesandes sõltub teie saadud punktide arv kodeeritud sõnumi ja esialgse sõnumi pikkuste suhtest.

Näide

Vaatleme juhtu, kus **N=3** ja **M=10,20,30**.

Meetod encode(N,M) võib mingit meile tundmatut viisi kasutades kodeerida sõnumi arvujärjendiks (7, 3, 2, 70, 15, 20, 3). Selle järjendi esitamiseks tuleb kutsuda meetodit send välja järgnevalt:

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

Oletame, et pärast kõikide papagoide sihtkohta jõudmist saime arvujärjendi (3, 20, 70, 15, 2, 3, 7). Meetod decode kutsutakse siis välja parameetritega **N=3**, **L=7** ja **X=3,20,70,15,2,3,7**.

Meetod decode peab taastama esialgse sõnumi (10, 30, 20) ja esitama tulemuse, kutsudes meetodit output välja järgnevalt:

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



Alamülesanded

Alamülesanne 1 (17 punkti)

- ✦ $N = 8$, ja iga täisarv massiivis M on kas 0 või 1.
- ✦ Iga kodeeritud arv peab olema lõigust 0 kuni $R=65535$.
- ✦ Meetodit send võite välja kutsuda ülimalt $K=10 \times N$ korda.

Alamülesanne 2 (17 punkti)

- ✦ $1 \leq N \leq 16$.
- ✦ Iga kodeeritud arv peab olema lõigust 0 kuni $R=65535$.
- ✦ Meetodit send võite välja kutsuda ülimalt $K=10 \times N$ korda.

Alamülesanne 3 (18 punkti)

- ✦ $1 \leq N \leq 16$.
- ✦ Iga kodeeritud arv peab olema lõigust 0 kuni $R=255$.
- ✦ Meetodit send võite välja kutsuda ülimalt $K=10 \times N$ korda.

Alamülesanne 4 (29 punkti)

- ✦ $1 \leq N \leq 32$.
- ✦ Iga kodeeritud arv peab olema lõigust 0 kuni $R=255$.
- ✦ Meetodit send võite välja kutsuda ülimalt $K=10 \times N$ korda.

Alamülesanne 5 (kuni 19 punkti)

- ✦ $16 \leq N \leq 64$.
- ✦ Iga kodeeritud arv peab olema lõigust 0 kuni $R=255$.
- ✦ Meetodit send võite välja kutsuda ülimalt $K=15 \times N$ korda.
- ✦ **NB:** Selles alamülesandes sõltub punktide arv kodeeritud sõnumi ja esialgse sõnumi pikkuste suhtest.

Olgu testi t korral $P_t = L_t / N_t$ kodeeritud sõnumi pikkuse L_t ja esialgse sõnumi pikkuse N_t suhe. Olgu P maksimum üle kõigi P_t . Saadavad punktid määratakse järgmise reeglite alusel:

- ✦ Kui $P \leq 5$, saate maksimaalsed 19 punkti.
- ✦ Kui $5 < P \leq 6$, saate 18 punkti.
- ✦ Kui $6 < P \leq 7$, saate 17 punkti.
- ✦ Kui $7 < P \leq 15$, saate $1 + 2 \times (15 - P)$ punkti, ümardatud allapoole lähima täisarvuni.
- ✦ Kui $P > 15$ või kui *mistahes* teie väljunditest on vigane, saate 0 punkti.
- ✦ **NB:** Alamülesannete 1 kuni 4 igasugune õige lahendus lahendab ka kõik eelnevad alamülesanded. Kuna alamülesandes 5 on lubatud K ülempiir suurem, ei tarvitse alamülesande 5 iga lahendus alamülesandeid 1 kuni 4 lahendada. Kõik alamülesanded on siiski võimalik lahendada ka ühe lahendusega.



Realisatsioon

Piirangud

- ✦ Hindamiskeskond: päris hindamiskeskonnas kompileeritakse teie programm kaheks programmiks **e** ja **d**, mis käivitatakse eraldi. Teie encoder ja decoder moodulid lingitakse mõlemasse programmi, aga **e** kutsub välja ainult encode ja **d** ainult decode meetodit.
- ✦ Protsessori ajalimiit: Programm **e** kutsub meetodit encode välja 50 korda ja see peaks lõpetama oma töö 2 sekundi jooksul. Programm **d** kutsub meetodit decode välja 50 korda ja see peaks lõpetama oma töö 2 sekundi jooksul.
- ✦ Mälulimiit: 256 MB.

Liides (API)

- ✦ Töökataloog: parrots/
- ✦ Võistleja lahenduse failid: encoder.cpp ja decoder.cpp
Märkus: nii näidishindajas kui ka päris hindajas lingitakse encoder.cpp ja decoder.cpp hindajaga. Seega peaksite kõik globaalsed muutujad mõlemas failis deklareerima võtmesõnaga static, et vältida segadusi muutujatega teistest failidest.
- ✦ Võistleja lahenduse liides: encoder.h ja decoder.h
- ✦ Hindaja liides: encoderlib.h ja decoderlib.h
- ✦ Testkeskkond: grader.cpp

Testkeskkond töötab kahes etapis. Kummaski etapis kutsub ta kõigepealt välja meetodi encode antud lähteandmetega ja seejärel kutsub ta välja meetodi decode teie meetodi encode väljundiga. Esimeses etapis testkeskkond kodeeritud sõnumis olevate arvude järjekorda ei muuda. Teises voorus vahetab testkeskkond paaritutel ja paariskohtadel olevad arvud. Tegelik hindamiskeskond rakendab kodeeritud sõnumitele erinevaid permutatsioone. Seda, kuidas testkeskkond andmeid segab, saate varieerida, muutes meetodit shuffle.

Testkeskkond kontrollib ka kodeeritud andmete väärtusi ja pikkust. Vaikimisi kontrollitakse, et kodeeritud andmed on lõigust **0** kuni **65535** ja pikkus oleks ülimalt $10 \times N$. Seda saate muuta, seadistades konstante **channel_range** (näiteks väärtusest 65535 väärtuseks 255) ja **max_expansion** (näiteks väärtusest 10 väärtuseks 15 või 7).

- ✦ Testkeskkonna sisendi näited: grader.in.1, grader.in.2

Märkus: Testkeskkonna sisendi vorming:

- ✦ Rida 1: N
- ✦ Rida 2: N arvust koosnev loend: $M[0], M[1], \dots, M[N-1]$
- ✦ Oma lahenduse testimiseks kompileerimine:
`g++ -o parrots encoder.cpp decoder.cpp grader.cpp`
- ✦ Oma lahenduse testimiseks käivitamine:
`cat grader.in.1 | ./parrots`