

Papagájok

Yanee papagájokkal akar üzenetet továbbítani, K papagája van. Egy M üzenet N darab, 0 és 255 közötti egész számokból álló sorozat. A papagájok 0 és R közötti egész számot tudnak megjegyezni és továbbítani.

Yanee azt gondolta, hogy az elküldendő M sorozat minden elemét betanítja egy-egy papagájjal, majd elküldi őket az üzenet céljához. A papagájok azonban nem abban a sorrendben érkeztek meg, ahogy küldte őket, ezért az üzenet összekeveredett.

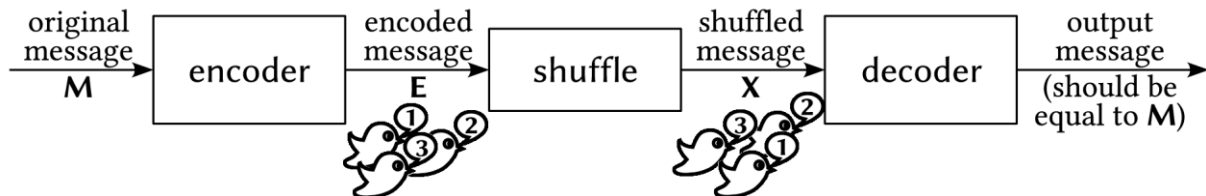
Olyan módszert kell alkalmazni, amely a madarak érkezési sorrendjétől függetlenül helyre tudja állítani a küldött üzenetet. Írj két eljárást:

- ♣ Az egyik az M üzenetet átalakítja legfeljebb K darab 0 és R közötti egész számok sorozatára.
- ♣ A másik a kapott 0 és R közötti egész számokból álló sorozatot átalakítja az eredeti M üzenetté.

Feladat

Írj két eljárást, az egyik a küldő (encoder), a másik a fogadó (decoder).

Az üzenetküldés és fogadás menete (shuffle jelentése sorrend átrendezés):



A két eljárás:

- ♣ `encode(N,M)`:
 - ♣ N – az üzenet hossza.
 - ♣ M – az üzenetet tartalmazó egydimenziós tömb, $0 \leq M[i] \leq 255$ ($0 \leq i < N$).

Ez az eljárás az M üzenetet 0 és R közötti egész számokból álló, legfeljebb K hosszú sorozattá alakítja, majd sorrendben a sorozat minden a elemére meghívja a `send(a)` eljárást.

Megjegyzés: R és K nem bemenő paraméter, lásd a tesztelésnél leírtakat!

- ♣ `decode(N,L,X)`:
 - ♣ N – az eredeti üzenet hossza.
 - ♣ L – a kapott üzenet hossza (a küldött madarak száma).
 - ♣ X – a kapott üzenetet, L darab egész számot tartalmazó tömb. Az $X[i]$ ($0 \leq i < L$) a `send` eljárás által elküldött számok valamelyike, de nem feltétlenül a megfelelő sorrendben.

Ez az eljárás határozza meg az eredeti üzenetet! Az `output(b)` eljárást kell hívni, az i . hívásban a b paraméter az eredeti üzenet i . eleme legyen!

Helyes megoldás esetén az alábbiaknak kell teljesülni:

- ⤴ Az encode által küldött számok korlátai a tesztelésben leírtak szerintiék.
- ⤴ Az encode a send eljárást legfeljebb K -szor hívja. Megjegyzendő, hogy K függ az üzenet hosszától.
- ⤴ A decode visszaállítja a teljes M üzenetet és az output(b) eljárást pontosan N -szer hívja, a hívásokban a b paraméter $M[0]$, $M[1]$, ..., $M[N-1]$, ebben a sorrendben.

Az utolsó tesztcsoportban a pontszám függ a kódolt és az eredeti üzenet hosszának arányától.

Példa

$N = 3$, és

10

$M =$ 30

20

Az encode(N, M), valamely módszert használva, az üzenetet a (7, 3, 2, 70, 15, 20, 3) sorozattal kódolja, ekkor a send hívásai:

send(7)

send(3)

send(2)

send(70)

send(15)

send(20)

send(3)

Ha a számok ilyen sorrendben érkeznek meg: $X = (3, 20, 70, 15, 2, 3, 7)$, $N = 3$, $L = 7$, akkor elő kell állítani az eredeti üzenetet: (10, 30, 20), tehát az output eljárást a következő 3 paraméterrel kell, ebben a sorrendben meghívni:

output(10)

output(30)

output(20)

Tesztek

1. eset (17 pont)

- ⤴ $N = 8$, és M minden eleme 0 vagy 1.
- ⤴ A kódolt egészek 0 és $R = 65535$ közöttiek lehetnek.
- ⤴ $K = 10 \times N$.

2. eset (17 pont)

- ⤴ $1 \leq N \leq 16$.
- ⤴ A kódolt egészek 0 és $R = 65535$ közöttiek lehetnek.
- ⤴ $K = 10 \times N$.

3.eset (18 pont)

- ⤴ $1 \leq N \leq 16$.
- ⤴ A kódolt egészek 0 és $R=255$ közöttiek lehetnek.
- ⤴ $K=10 \times N$.

4. eset (29 pont)

- ⤴ $1 \leq N \leq 32$.
- ⤴ A kódolt egészek 0 és $R=255$ közöttiek lehetnek.
- ⤴ $K=10 \times N$.

5. eset (legfeljebb 19 pont)

- ⤴ $16 \leq N \leq 64$.
- ⤴ A kódolt egészek 0 és $R=255$ közöttiek lehetnek.
- ⤴ $K=15 \times N$.
- ⤴ **Fontos:** a pontszám függ a kódolt és az eredeti üzenet hosszának arányától.
A t . tesztszenben $P_t=L_t/N_t$ (ahol L_t kódolt üzenet hossza, N_t pedig az eredeti üzenet hossza). Jelölje P a P_t értékek maximumát! Pontszámot az alábbiak szerint kapsz:
 - ⤴ Ha $P \leq 5$, akkor **19** pont.
 - ⤴ Ha $5 < P \leq 6$, akkor **18** pont.
 - ⤴ Ha $6 < P \leq 7$, akkor **17** pont.
 - ⤴ Ha $7 < P \leq 15$, akkor az $1 + 2 \times (15 - P)$, lefelé kerekítve.
 - ⤴ Ha $P > 15$ vagy hibás esetben, a pontszámod **0**.
- ⤴ **Fontos:** Az 1..4 tesztcsoporthoz bármelyikére adott helyes megoldás az összes korábbinak is helyes megoldása. Az 5. tesztcsoporthoz ez nem biztos, hogy igaz a magasabb K korlát miatt.

Határok

- ⤴ Értékelő rendszer két programot készít (**e** és **d**), amelyeket külön futtat. Az encode és a decode eljárást mindkettőhöz hozzászerkesztik, de mindegyik csak a megfelelőt hívja meg.
 - ⤴ Időlimit: Az **e** 50 encode hívást végez, időlimitje 2 másodperc. A **d** 50 decode hívást végez, időlimitje 2 másodperc.
 - ⤴ Memórialimit: 256 MB
- Megjegyzés:** A verem méretre nincs külön korlát.

Interfész (API)

- ⤴ A megoldás könyvtára: parrots/
- ⤴ A megvalósítandó modul:
 - ⤴ encoder.c vagy encoder.cpp vagy encoder.pas
 - ⤴ decoder.c vagy decoder.cpp vagy decoder.pas

C/C++ programozóknak: a minta és a valódi értékelő is hozzászerkeszti az encoder.c[pp] és a decoder.c[pp] modulokat az értékelőhöz. Ezért a modulban deklarált változókat globális változóknak kell deklarálni (például: global int n)!

-
- ⤴ Saját interfész:
 - ⤴ encoder.h vagy encoder.pas
 - ⤴ decoder.h vagy decoder.pas
 - ⤴ Értékelő interfész:
 - ⤴ encoderlib.h vagy encoderlib.pas
 - ⤴ decoderlib.h vagy decoderlib.pas
 - ⤴ Minta értékelő : grader.c vagy grader.cpp vagy grader.pas

A mintaértékelő kétszer hívja az eljárásaidat. Először az üzenet sorrendjét nem változtatja meg, másodszer pedig felcseréli a páros és páratlan sorszámú elemeket. Ha más permutációval is akarod tesztelni a megoldásodat, akkor a shuffle (C/C++) vagy a Shuffle (Pascal) eljárást módosítsd! A valódi értékelő sokféleképpen permutálhatja az üzenet elemeit.

A mintaértékelő az R és a K korlátokat is teszteli, R=**65535**, K=10×N. Ha más kell, változtasd meg a **channel_range** (pl. 65535-ről 255-re) és **max_expansion** (pl. 10-ről 15-re) konstansokat!

- ⤴ Minta bemenetek: grader.in.1, grader.in.2, ...

Megjegyzés: A minta értékelő a bemenetet a következő formában olvassa:

- ⤴ 1. sor: N
- ⤴ 2. sor: M[0], M[1], ..., M[N-1]

- ⤴ A minta bemenetre elvárt kimenetek: grader.expect.1, grader.expect.2, ..., amelyekben a „Correct.” szövegnek kell lenni.