

Планина

Ави е на поход по планинска пътека, разделена на N участъка. Надморските височини на участъците се характеризират от редицата a_0, a_1, \dots, a_{N-1} . Макар и да е в добра физическа форма, Ави е поела по полегата пътека - такава, че разликите във височините на всеки два съседни участъка не надвишава 1, т.е. $|a_{i+1} - a_i| \leq 1$ за всяко $0 \leq i \leq N - 2$.

Момичето иска да изпрати на приятеля си Боби редицата a . Ако сте ходили на планина обаче, ще знаете, че обхватът там (и съответно мобилният интернет на Ави) е ужасен. Поради тази причина тя не може да изпрати повече от M_{max} бита на приятеля си.

Вашата задача е да напишете *две програми*, чрез които Ави да може да изпрати редицата a на Боби, използвайки най-много M_{max} бита.

- Първата програма получава редицата a и я кодира в M бита b_0, b_1, \dots, b_{M-1} .
- Втората програма получава генерираните от първата програма битове b и дължината на пътеката N и трябва да възстанови оригиналната редица a .

Детайли по имплементацията

Трябва да изпратите два сорс файла `avi.cpp` и `bob.cpp`, реализиращи двете програми.

Освен долните функции, Вашите програми могат да ползват и реализират други вътрешни функции и глобални променливи. Тъй като изпратените от Вас два сорс файла ще бъдат компилирани заедно с грейдър в една програма, тези вътрешни функции и глобални променливи трябва да бъдат деклариран с ключовата дума `static`.

За всеки тест така компилираната програма ще бъде изпълнена два пъти - веднъж изпълнявайки програмата на Ави и веднъж тази на Боби.

`avi.cpp`

В началото си Вашата програма `avi.cpp` трябва да съдържа `#include "avi.h"`. Вашата програма `avi.cpp` трябва да реализира следната функция:

- `std::vector<bool> encode(std::vector<int> a, int m_max)`
 - a : редицата от височините на участъците на пътеката
 - m_max : максималният брой битове M_{max} , които Ави може да изпрати на Боби.
 - Функцията трябва да връща редица b с дължина $M \leq M_{max}$ - съобщението, което Ави изпраща на Боби.
 - Функцията се вика веднъж за всеки тест.

В случай, че вашата функция `encode` върне редица с повече от M_{max} бита, ще получите съобщение „Output isn't correct.“.

bobi.cpp

В началото си Вашата програма `bobi.cpp` трябва да съдържа `#include "bobi.h"`.
Вашата програма `bobi.cpp` трябва да реализира следната функция:

- `std::vector<int> decode(std::vector<bool> b, int N)`
 - `b`: съобщението, чрез което вашата програма `avi.cpp` е кодирала `a`
 - `N`: броя участъци в пътеката.
 - Функцията трябва да връща редица с дължина `N` - височините на участъците на пътеката, които е получила вашата програма `avi.cpp`.
 - Функцията се вика веднъж за всеки тест.

В случай, че вашата функция `decode` не върне редица `a`, различна от зададената, ще получите съобщение „Output isn't correct.“.

Пример

Грейдърът извършва следното извикване на функция от `avi.cpp`:

- `encode([0, 1, 1, 0, 1], 20)` В случая $a_i \leq 1$ за всяко $0 \leq i \leq N - 1$ и всяка височина може да се запише с един бит. Функцията връща редицата `1, 0, 0, 1, 0`.

След това грейдърът се пуска наново и извършва следното извикване на функция от `bobi.cpp`:

- `decode([1, 0, 0, 1, 0], 5)` Функцията трябва да върне `0, 1, 1, 0, 1` - оригиналната редица от височини `a`.

Подзадачи

Във всички подзадачи $0 \leq a_i \leq 10^9$ за всяко $0 \leq i \leq N - 1$.

1. (25 точки) $1 \leq N \leq 1000$, $M_{max} = 200\,000$
2. (25 точки) $1 \leq N \leq 100\,000$, $M_{max} = 200\,000$
3. (25 точки) $1 \leq N \leq 100\,000$, $M_{max} = 170\,000$, $|a_{i+1} - a_i| = 1$ за всяко $0 \leq i \leq N - 2$.
4. (25 точки) $1 \leq N \leq 100\,000$, $M_{max} = 170\,000$

Локално тестване

За да можете да тествате решението си на компютъра си, Ви се предоставят файловете `Lgrader.cpp`, `avi.h` и `bobi.h`, които да компилирате заедно с Вашите сорсове `avi.cpp` и `bobi.cpp`.

За разлика от грейдъра на системата, локалният грейдър изпълнява функциите `encode` и `decode` наведнъж.

Вход

- Ред 1: две цели числа N и M_{max}
- Ред 2: N числа a_0, a_1, \dots, a_{N-1}

Изход

- Ред 1: редицата `b`, резултатът от Вашата функция `encode`

- Ред 2: Нека означим дължината на b с M .
 - Ако $M \leq M_{max}$, редица от N числа, резултатът от Вашата функция `decode`
 - Ако $M > M_{max}$, „Too many bits used“.

Изпращане на тестове към системата

За съжаление изпращането на тестове към системата не работи в момента.