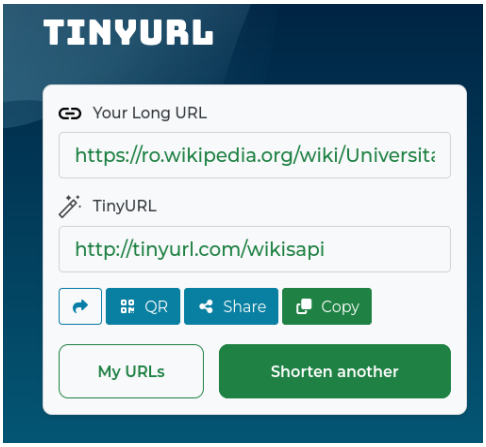
PARCIÁLIS II. - LINKRÖVIDÍTÉS



Bizonyára már használtatok olyan szolgáltatásokat, amely segítségével hosszabb linkekből képesek voltatok rövidebb linkeket generálni.

A működési elvük nagyon egyszerű: egy háttér tárolóban lementik az eredeti, hosszú elérhetőségi vonalat, és hozzátársítanak egy rövidebb, könnyebben megjegyezhető címkét. A feladatban felmutatott példában rövidítésre kerül a https://ro.wikipedia.org/wiki/Universitatea_Sapientia link, amely a következő, rövidebb linken keresztül elérhetővé válik: https://tinyurl.com/wikisapi

Feladatod, hogy létrehozd saját linkrövidítő szolgáltatásod!

A linkrövidítő szolgáltatás segítségével, amennyiben felmutatják a felhasználók a saját API kulcsukat, olyan rövid linket tudnak generálni, amelyekhez hozzárendelhetőek engedélyezett API kulcsok. A rövid linkeket csak egy engedélyezett API kulcs segítségével lehet feloldani hosszabb linkekre. Alapértelmezetten az API kulcs, amellyel létrehoztuk a rövid linket, fel tudja oldani a saját linkjét! A rövid linkek egyediek, nem generálható ki ugyanaz a rövid link kétszer (vigyázat: két rövid link attól még képes ugyanarra a hosszú linkre mutatni)!

- Példa egy hosszú linkre: **https://ro.wikipedia.org/wiki/Universitatea_Sapientia**
- Példa egy rövid linkre: **https://sapi.ro/abcdEFG**
- Példa egy rövid link egyedi részére**: abcdEFG**
- Példa egy API kulcsra: **4f726498-924a-46d6-bdc4-77f4e41bfb9f**
- Példa bemeneti állományra:

```
https://www.youtube.com/watch?v=8Wh8FaZmAVQ 3
bb702ffd-640d-40fe-8520-94bac77c30aa
7f771d3a-6c5a-476e-913a-b5a9d574e095
abe059aa-acb9-42fb-9dee-0411b4d5a1fe
https://reddit.com/r/Python/comments/18ufktx 1
0d537dc4-7bc2-4689-be5a-47f7b2996523
https://cdn.mos.cms.futurecdn.net/iC7HBvohbJqExqvbKcV3pP.jpg 2
a6f19f7d-cf6b-4c87-85b7-0c36d42c927f
f81effdb-85b9-4243-91ac-4dbade228abe
```

(1p) Hozz létre egy új osztályt, amely a rövidített linket képviseli. Tartalmazzon minden olyan adattagot, amely releváns egy rövidített link modellezéséhez, illetve azt is rögzítse, hogy hányszor oldották fel eddig. Az adatokat a memóriában tároljuk!

- **Ötletek**: eredeti, hosszú link tárolása, feloldások száma, engedélyezett API kulcsok tárolása valamely módon. Tárolható még egyszer a rövid link is.
- **Megkötés**: végtelen mennyiségű API kulcsot lehessen hozzárendelni egy rövid linkhez, ne használjunk kötött méretű tárolókat!
- **Ajánlás**: Nem szükséges a teljes rövid linket eltárolni, elég, ha csak az egyedi részt tároljuk: például **https://sapi.ro/abcdEFG** esetében **abcdEFG**

(1p) Az osztályban hozd létre a következő függvények leírását:

1. Legyen egy függvény, amely visszatéríti a hosszú linket;
2. Legyen egy függvény, amely visszatéríti azokat az API kulcsokat, amelyek hozzáférnek a rövidített linkhez. Nem szükséges az adatok biztonsági másolása.
3. Legyen egy függvény, amely segítségével megvizsgálhatjuk, ha egy adott API kulcs hozzáfér-e a rövidített linkhez;
4. Legyen egy függvény, amely segítségével hozzárendelhetünk egy új API kulcsot a rövidített linkhez.

(2p) Implementáld az osztály függvényeit!

- **Megkötés:** Használd az STL által rendelkezésre bocsátott kereső függvényeket, ne implementáld le a keresést egyszerű for ciklussal **(0.5p)**
- **Vigyázat!** Ha már hozzá van rendelve egy API kulcs a rövidített linkhez, ne engedjük azt mégegyszer hozzárendelni! Ebben az esetben dobjunk kivételt: `std::invalid_argument` **(0.5p)**

(1p) Hozz létre egy segédfüggvényt, amely kigenerálja egy véletlenszerű rövid link egyedi részét (például: abcdEFG). Bármilyen stratégiát használhatsz, amennyiben elegendően véletlenszerű linkeket generál!

- **Ajánlás:** használd a C++ által nyújtott véletlenszerű számgenerálást:

```
std::random_device dev;
std::mt19937 rng(dev());
std::uniform_int_distribution<int> numbers(1, 26); // [1, 26] tartományból
generál számokat

int number = numbers(rng);
```

(1p) Hozz létre egy interfészt (C++-ban: absztrakt osztály tiszta virtuális függvényekkel), amely leírja a linkrövidítő funkcionalitásait:

1. Legyen képes egy hosszú linkből rövid linket készíteni, ha megadjuk a felhasználó API kulcsát. Térítse vissza a rövid link egyedi részét!
2. Legyen képes egy rövidített linket feloldani egy hosszabb linkre, amennyiben megadunk egy olyan API kulcsot, amely hozzáféréssel rendelkezik a rövidített linkhez. Térítse vissza a hosszú linket, vagy dobjon kivételt (amennyiben nem létezik a rövidített link, vagy nem jogosult a feloldásra a felmutatott kulcs)!
3. Legyen képes egy rövidített linkhez hozzárendelni egy új, engedélyezett API kulcsot, amennyiben megadjuk a rövidített link egyedi részét és az új API kulcsot. Dobjon kivételt (amennyiben nem létezik a rövidített link, vagy már hozzá van rendelve a kért kulcs a linkhez)!
4. Legyen képes egy bemeneti állományból kiolvasni az összes hosszú linket a memóriába és hozzárendelni a megfelelő API kulcsokat, amennyiben megadjuk a bemeneti állomány nevét.

(2p) Implementáld az előző alpontban leírt interfészt egy Service osztályon belül!

(1p) Teszteld le alaposan a megírt program összes függvényét a `main.cpp` állományban.