Tanári útmutató

Kódok a hétköznapokban

Babus Réka, Juhász Péter

I. ÁTTEKINTÉS

A tanóra során szeretnénk bemutatni mindennapjaink néhány olyan területét, ahol rendszeresen találkozhatunk különböző számkódok használatával: szó lesz a színkódokról, a vonalkódokról, valamint említés szintjén a QR-kódokról is.

A hexadecimális színkódokat sokan ismerik, ám feltételezhető, hogy a legtöbben csak használják ezeket, anélkül hogy pontosan értenék, miért éppen azokból a karakterekből állnak, amelyekből.

Vonalkódokkal és QR-kódokkal is nap mint nap találkozunk, működésük alapelveivel azonban valószínűleg kevesen vannak tisztában.

A tanóra kitér arra is, hogy milyen egyszerűbb matematikai módszereket alkalmaznak a különböző kódok hibáinak kiszűrésére és javítására. Míg a QR-kódok esetében ez a téma meghaladja a középiskolai szintet, a vonalkódok (illetve a bankszámlaszámok, ISBN-számok, személyi azonosítók stb.) esetében a működési elv könnyen megérthető.

Reméljük, hogy ezzel sikerül inspirálni a diákokat arra, hogy ne vegyék természetesnek a körülöttünk lévő világot, hanem igyekezzenek azt megérteni, sőt, esetenként megkérdőjelezni is.

II. A tanóra egyes elemeinek időbeli becslése

❗ Fontos: Az egyes egységekhez rendelt számok reméljük, hogy segítenek megbecsülni, hogy az egyes feladatok hány percet tesznek ki. Ezek viszont csak irányadóak, érdemes a diákok tempójához igazítani, hogy végül mire, mennyi időt szánunk.

Az óra elemei	Rövid leírás	Módszer	Várható időtartam
1.	A hexadecimális színkód működése	kérdezve kifejtés, tanári előadás	12 perc
2.	Színkitalálós játék	önálló munka	12 perc
3.	A vonalkódok működése	kérdezve kifejtés, tanári előadás	10 perc
4.	Hibajavító algoritmus vonalkódoknál	kérdezve kifejtés, tanári előadás	11 perc

A prezentáció kivetítéséhez szükség lesz laptopra és projektorra. Fontos, hogy a gyerekeknél legyen legalább párokban egy telefon vagy okoseszköz! Ezen kívül csak tábla és filc szükséges a tanórához.

III. A TANÓRA MENETE

1. A hexadecimális színkód működése

Érdekes bevezetés lehet, ha a diákok megpróbálják megfejteni, mit jelenthetnek a diasor első diáján látható kódok. Valószínű, hogy lesz, aki rögtön felismeri a hexadecimális színkódokat, és olyan is akad, aki már hallott ezek működéséről. Praktikus, ha először a diákok gyűjtik össze, hogy mit tudnak a témáról.

Ezt követően mi is foglaljuk össze a színkódok működésének lényegét:

A hatjegyű kód három darab, 16-os számrendszerbeli, kétjegyű számból tevődik össze. A színeket úgy definiáljuk, hogy azok előállításához mennyi piros, zöld és kék fényre van szükség. Az első kétjegyű szám adja meg a **piros** komponens intenzitását, a második a **zöldét**, a harmadik pedig a **kékét**.

Több tanuló számára újdonság lehet a 16-os számrendszer. Érdemes először rákérdezni, hogy milyen számjegyeket tartalmaz, majd célszerű a táblára felírni az **A, B, C, D, E, F** karakterek értékét is.

Ezt követően hasznos lehet két rövid átváltási feladatot megoldani, hogy a tanulók gyakorolhassák a 10-es és 16-os számrendszer közötti átváltást.

Váltsd át 10-es számrendszerre az **#FBB42F** kódot!

💡 Megj.: Itt jó, ha hangsúlyozzuk, hogy a 10-es számrendszerbeli átváltás jelen esetben azt takarja, hogy három darab tízes számrendszerbeli 2-jegyű számra szeretnénk a karaktersorozatot átváltani.

Megoldás:
Az első kétjegyű szám az FB, ami $11 \cdot 1 + 15 \cdot 16 = 251$-gyel egyenlő.
A második kétjegyű szám a B4, ami $4 \cdot 1 + 11 \cdot 16 = 180$-nal egyenlő.
A harmadik kétjegyű szám a 2F, ami a $2 \cdot 1 + 15 \cdot 16 = 242$-vel egyenlő.

Mi lesz a (255,32,180) hexadecimális kódja?

Megoldás:

$255 = 15 \cdot 16 + 15 \cdot 1$, így a kód FF. $32 = 2 \cdot 16$, így ennek a kódja 20. $180 = 11 \cdot 16 + 4 \cdot 1$, így ennek a kódja B4. A hex kód tehát: #FF20B4

💡 Megj.: A hex kód a hexadecimális kód rövid elnevezése.

Ezen a ponton a diákok remélhetőleg már értik, hogyan válthatók át a hexadecimális színkódok a 10-es számrendszerbeli számpárokra, és tudják, hogy ezek a piros, zöld és kék komponensek intenzitását jelölik. Az azonban még nem feltétlenül egyértelmű, hogy az egyes komponensek mennyisége miként befolyásolja a végső színt. Ahhoz, hogy ezt jobban megértsük, érdemes néhány szót ejteni **az additív színkeverésről**.

Az **additív színkeverés** a fények keverésének elvén alapuló színképzési mód, amely a kijelzők, monitorok, projektorok, okostelefonok és televíziók működésének alapját képezik. Lényege, hogy különböző színű fényforrások összeadásával (vagyis egymásra vetítésével) új színek jönnek létre. Alapszínei a piros, zöld és kék; ezek különböző arányú keverésével minden más szín előállítható.

Ha egyik fény sincs jelen, fekete színt (a fény hiányát) látjuk – hiszen a kijelzők alapszíne a fekete. A fekete kódja tehát: #000000. Minél több színt adunk hozzá a keverékhez, annál világosabb árnyalat keletkezik. Fehér színt akkor kapunk, ha mindhárom fényt azonos, maximális intenzitással vetítjük egymásra. Ennek kódja: #FFFFFF.

A szemünk nem egyformán érzékeli a színeket: a zöldre érzékenyebb, mint a vörösre, és a vörösre érzékenyebb, mint a kékre, ezért a fényerő (luminancia) nem egyszerűen az RGB-értékek átlaga. Erről további információ itt található:

💡 Megj.: Gyakori színkeverés még a **szubtraktív.** Ez a festékek, tinták vagy pigmentek keverésének elvén alapul, nyomtatáskor ezt használjuk. A fehér az, ami minden ráeső fényt visszaver, ez a kiindulási szín, a papír színe. Szubtraktív színkeverésnél a visszavert fényből veszünk el színeket, hogy valamennyi fényt elnyeljen. Alapszínei a cián, sárga és a magenta.

2. Színkitalálós játék

Bár a fényerő meghatározása összetett folyamat, ha látjuk, hogy bizonyos színek keveréke milyen árnyalatot eredményez, viszonylag pontosan meg tudjuk határozni, hogy egy adott kód milyen színt takar – és fordítva.
Nézzünk rá a színkörre, hogy erről pontosabb képet kapjunk, majd próbáljuk ki, mennyire tudjuk az eddig tanultak alapján meghatározni a különböző színek kódját!

Nyissuk meg a következő játékot:
https://rekababuska.github.io/orakakonyvbol/szamrendszerek/szinkod.html

Hogy ugyanazt a színt kapja minden tanuló, írja be mindenki ugyanazt a tetszőleges számkódot. (Például mindenki a 34-est.) A tanulók feladata, hogy megtippeljék a generált szín színkódját. A tipp beküldése után a weboldal megmutatja a kóddal megadott színt és azt is, hogy a tipp és az eredeti szín egymástól milyen távolságra vannak.

Az új ismeretek fényében a diákoknak lehetősége van kétszer módosítani a tippjüket azzal a céllal, hogy a végén minél *közelebb* kerüljenek a megadott színhez. A nyertes az, aki 3 tippel a lehető legközelebb kerül az eredeti színkódhoz.

Opcionális kiegészítés

Erősebb csoportoknál adja magát, hogy beszéljünk arról is, mit jelent a honlapon megjelenő „távolság” érték, és hogyan számítódik ki. Rávezető kérdésként feltehetjük: *Hogyan lehetne ezeket a három komponensből (piros, zöld, kék) álló kódokat ábrázolni?*

Remélhetőleg lesz olyan diák, aki javasolja, hogy tekintsük ezeket egy háromdimenziós koordinátarendszer pontjaiként, ahol az RGB-értékek adják a koordinátákat.
A két pont közötti távolságot ekkor a háromdimenziós Pitagorasz-tétel segítségével számíthatjuk ki. Úgy is szemlélhetjük, hogy a két pont távolsága egy téglatest testátlójának hossza – ezzel az összefüggéssel a tanulók valószínűleg már találkoztak korábban.

💡 Megj.: A weboldalon található egy másik játék is, ahol egy adott kódhoz kell színt választani. Ezt opcionális házi feladatnak javasoljuk, hogy elég idő maradjon a további feladatokra.

3. A vonalkódok működése

Bár a vonalkódok nem egy konkrét számrendszerre épülnek, működésük mégis érdekes, és jól kapcsolódik a számrendszerek tágabb alkalmazásaihoz.

A témát azzal vezethetjük be, hogy megvizsgáljuk, milyen igény hívta életre a vonalkódok létrejöttét: a bolti eladók korábban kézzel írták be a termékek árát a számítógépbe, ami sok időt vett igénybe, és sok hibával is járt. Szükség volt tehát egy gyorsabb, automatizált azonosítási módszerre.

A vonalkódra tekinthetünk úgy, mint egy morzekódra, ahol a rövid és hosszú jelek helyett függőleges fekete és fehér téglalapok jelennek meg.
A fekete csíkok elnyelik, a fehérek pedig visszaverik a fényt. A fény visszaverődését egy érzékelő méri:
– ahonnan sok fény jut vissza, ott fehér sáv van,
– ahonnan kevés, ott fekete.
A világos–sötét mintázatok alapján a készülék dekódolja a számokat.

A vonalkódok elején, közepén és végén két-két függőleges csík található, amelyek a szerkezetet tagolják. A legbaloldalibb szám a termék típusára utal, a bal oldali további öt számjegy a gyártóról ad információt, a jobb oldali öt számjegy a terméket azonosítja, a legutolsó szám pedig egy ellenőrzőszám, amelyről rövidesen szó lesz.
Így összesen 12 számjegyből áll egy standard (UPC) vonalkód.

💡 Megj.: Az UPC mellett gyakori az EAN-13 kód is, ami 13 számjegyből áll.

Minden számjegy 7 fekete vagy fehér téglalapból épül fel, vagyis a teljes vonalkód 12 darab 7-es blokkból áll. Ezen a ponton adja magát a kérdés, hogy vajon reális veszély-e az, hogy kifogyunk a vonalkódokból.

Hányféle fekete-fehér kombináció készíthető?

Megoldás:

Kétféle megközelítésre számíthatunk. Az egyik az, hogy a ténylegesen létező vonalkódokat számolják a tanulók. Ennek száma 10^{12}, hiszen minden számjegy 10 féle lehet.

Feltehetően lesznek olyanok akik a fekete-fehér téglalap kombinációkból készíthető kódokat számolják meg. Ezzel a gondolatmenettel: egy blokk 7 fekete/fehér téglalapból áll. Így ez alapján egy blokk 2^7 féle lehet. A 12 blokk mindegyike 2^7 féle lehet, így összesen $(2^7)^{12} = 2^{84}$ lehetőségünk van.

Ha előkerül mindkét gondolatmenet, akkor érdemes összehasonlítani a két szám nagyságrendjét, ezáltal pontosabb képet kaphatunk arról is, hogy a 2^{84} mekkora számot takar.

A feladat megbeszélése után vizsgáljuk meg, hogy az egyes számjegyeket milyen fekete–fehér kombinációk reprezentálják. Látható, hogy ezek a kombinációk nem a kettes számrendszer alapján lettek meghatározva. Érdemes feltenni a kérdést: *vajon milyen logika szerint rendelik hozzá a fekete–fehér mintázatokat a számokhoz?*

Remélhetőleg lesz olyan a diákok között, aki rájön, hogy mindez azzal a céllal történt, hogy a kódok minél különbözőbbek legyenek egymástól, és így nehezen lehessen őket összekeverni a beolvasás során. Érdekes megfigyelni, hogy a bal és jobb oldalon a fekete és fehér téglalapok szerepe felcserélődik:
– a bal oldalon a fehér a 0, a fekete az 1,
– a jobb oldalon ez pont fordítva van.

Továbbá észrevehetjük, hogy a bal oldali kódokban mindenhol páros számú 0 és páratlan számú 1 szerepel. Ebből hamar rá lehet jönni, hogy a jobb oldalon ezzel szemben páratlan számú 0 és páros számú 1 található. (Ha az idő engedi, akkor érdemes őket megkérdezni, hogy milyen hasonlóságot látnak a bal oldali kódokban?)

Ezen a ponton érdemes feltenni a kérdést: *Vajon miért alakították ezt így ki?*

Valószínűleg lesz olyan, aki rájön, hogy mindennek az a célja, hogy a vonalkód fejjel lefelé is olvasható legyen – ezzel a kódolással a leolvasó mindig el tudja dönteni, honnan kezdődik a kód.

Hibajavító algoritmus vonalkódoknál

Korábban említettük, hogy az utolsó számjegy szerepére még visszatérünk. Ennek a számnak az a funkciója, hogy ha a vonalkód bármely része nem olvasható be vagy hibás, akkor is kiolvasható legyen a teljes kód. Nézzük meg, hogy mindez milyen algoritmus szerint működik:

1. Összeadjuk a páratlan helyen álló számjegyeket.
2. Az összeget 3-mal szorozzuk.
3. Hozzáadjuk a páros helyen álló számjegyek összegét.
4. Az ellenőrzőszám az az érték lesz, ami az összeget 10-zel oszthatóra egészíti ki.

💡 Megj.: Az ok, hogy az ellenőrzőszám nem egyszerűen csak az összeg 10-es maradéka az, hogy így nem csak egy számjegy hiányát, hanem két számjegy felcserélését is ki lehet szűrni.

Ezen a ponton praktikus lehet a dián látható vonalkódon végignézni az algoritmus lépéseit, és megállapítani, hogy valóban 4 lesz az ellenőrzőszám.

Vonalkód kiegészítése

A fenti algoritmus segít abban, hogy ha a vonalkód valamelyik része hiányzik, azt ki tudjuk egészíteni. Ezt kipróbálhatjuk az alábbi játékkal:

https://rekababuska.github.io/orakakonyvbol/szamrendszerek/vonalkod.html

Annak érdekében, hogy minden diák ugyanazt a feladványt kapja, mindenki írja be ugyanazt a tetszőleges számkódot (1–100 között).

A feladat menete:

1. a vonalkódnak valamelyik része hiányzik
2. az ellenőrző szám segítségével ki kell találni, mi a hiányzó számjegy
3. ezután meg kell találni a hiányzó számjegyhez tartozó fekete–fehér téglalap kombinációt.

Érdemes a dián kivetítve hagyni, hogy melyik számhoz milyen kód tartozik. Beugratós, hogy ha a vonalkód jobb oldaláról hiányzik a részlet, akkor a dián megadott kódokat pont fordítva kell értelmezni. Ami fehér volt az fekete lesz, és viszont.

Ezzel a logikával kiválasztható a hiányzó részlet.

Lezárásnak javasoljuk, hogy mutassuk meg a diasorban mellékelt képet egy QR kódról, aminek a belseje hiányzik. Meglepő, hogy így is be tudjuk olvasni a kódot és meg tudjuk nyitni a honlapot, mely Andrea Bocelli - Time To Say Goodbye klasszikusára visz, ezzel jelezve az óra végét. Érdemes megemlíteni, hogy itt is hasonló, de az előzőnél jóval összetettebb hibajavító algoritmusok vannak a háttérben.

IV. Opcionális kiegészítések

Az óraterv összeállítása során több olyan kapcsolódó témakörrel is találkoztunk, amelyet érdekesnek találtunk, de időhiány miatt végül nem került be az óravázlatba. Ezeket gyűjtöttük itt össze. Van köztük olyan, amely házi feladatként vagy óra eleji bemelegítő játékként is használható, de akad olyan is, amely jóval több időt igényel. Ha a megtartott óra után szívesen foglalkoznátok még a témával, bátran válogassatok az itt található feladatok közül.

QR kódok

A vonalkódok témakörének kézenfekvő folytatása a QR-kód. Érdekes lehet röviden beszélni a kialakulásukról, valamint arról, hogy milyen részekből állnak és hogyan működnek. A QR-kódok már jóval összetettebbek a hagyományos vonalkódoknál, ezért nem tartjuk reális célnak, hogy a tanulók minden részletét megértsék. Ugyanakkor az is hasznos lehet, ha a működési mechanizmusukról kapnak egy átfogó képet.

A QR kódok létrejöttéről

Az első QR kódot Masahiro Hara, a Denso Wave mérnöke fejlesztette ki 1994-ben. Célja az volt, hogy az autógyártásban használt alkatrészeket gyorsabban és több adattal lehessen azonosítani, mint a hagyományos vonalkódokkal. Ezzel egyidejűleg a kergemarhakór járvány (az 1990-es években) rávilágított arra, mennyire fontos az élelmiszerek — különösen a húsok — eredetének pontos nyomon követése. A fogyasztók és hatóságok szerették volna tudni, honnan származik a termék, és hogyan jutott el a boltokba. A QR-kód (Quick Response code) pont erre a célra bizonyult hasznosnak: sok információt tud elraktározni kis helyen, és gyorsan olvasható géppel. Masahiro Hara és a Denso Wave úgy döntöttek, hogy nem szabadalmaztatják a QR-kód technológiát, hanem ingyenesen elérhetővé teszik bárki számára. A kódot a GO táblajáték ihlette.

A QR kódok részei, működése

Az erre vonatkozó dia alján található egy QR kód, ami az alábbi youtube videóra vezet: https://www.youtube.com/watch?v=w5ebcowAJD8 A videó alaposan, érthetően bemutatja a QR kódok részeit, és működését. Az érdeklődő gyerekeknek lehet ajánlani, hogy nézzék meg.

Bár nap mint nap találkozunk QR kódokkal, talán nem tudatosítjuk magunkban, hogy mindegyikben vannak közös elemek. A kódok nemcsak adatokat tartalmaznak, hanem olyan mintákat is, amelyek segítik a leolvasást, a tájolást és a hibajavítást.

Az egyes részek funkciói röviden a következők:

1. Pozíciómeghatározó minták
   A három nagy, négyzet alakú minta a sarkokban található. Ezek segítik a szkenner programot, hogy felismerje a QR-kód helyzetét és irányát.

2. Igazító minták
   Kisebb négyzetek, amelyek a kód közepén és más pontjain találhatók. Feladatuk, hogy a torzításokat (például a ferde szkennelést) korrigálják.

3. Időzítési minták
   Váltakozó fekete-fehér négyzetek sora a pozícióminták között. Segítenek meghatározni az adatterület celláinak méretét és elhelyezkedését.

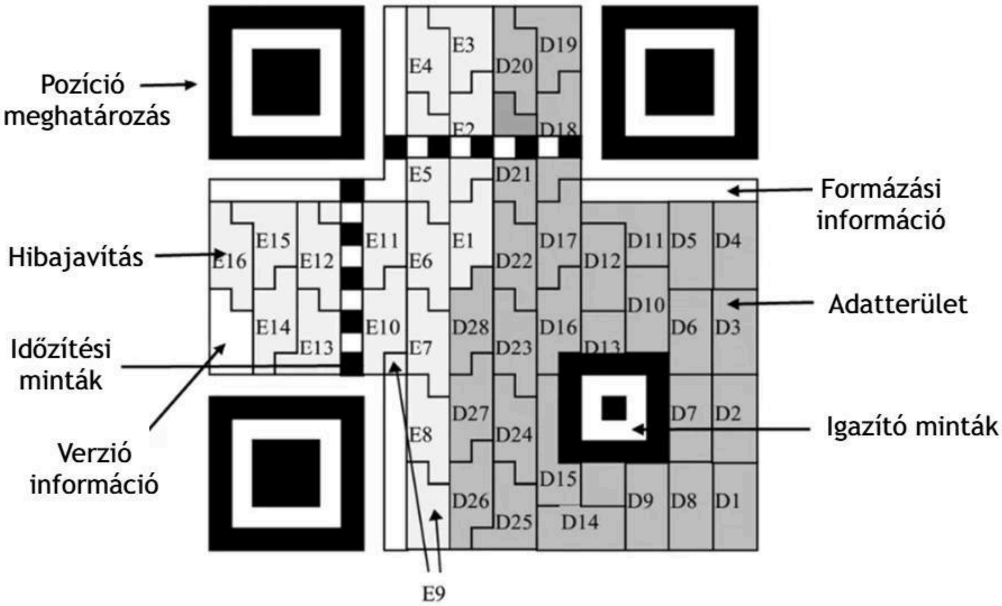
4. Verzió információ
   A QR-kód verziójáról (méretéről, komplexitásáról) ad információt. Nagyobb QR-kódoknál jelenik meg.

5. Formázási információ
   A hibajavítás szintjét és az adatmaszkolási mintát tartalmazza, amely befolyásolja, hogyan kell az adatokat értelmezni.

6. Hibajavító kód (Error correction)
   Olyan adatrészek, amelyek segítségével a QR-kód akkor is olvasható marad, ha részben sérült vagy szennyezett.

7. Adatterület

   Itt találhatók a tényleges információk (pl. szöveg, link, számok), bináris formában kódolva.



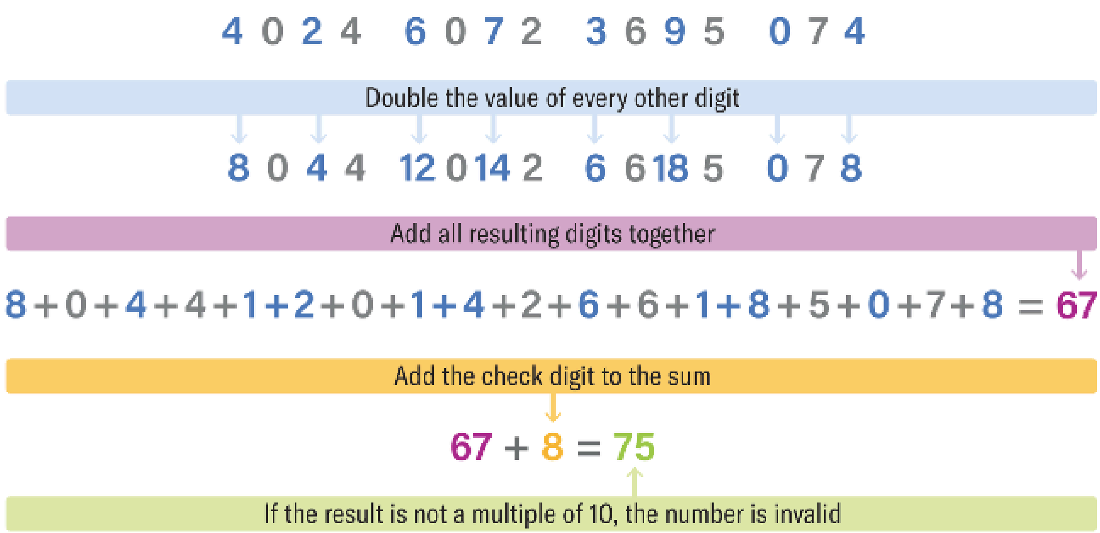
💡 Megj.: A videóban ezekről is további információkat találhatunk.

További hibajavító javító algoritmusok

A vonalkódok mellett a mindennapokban több más területen is találkozhatunk hibajavító algoritmusokkal. Ezekre hozunk néhány egyszerűbb példát. Az algoritmusok működése sok szempontból hasonló, ezért ha valaki szeretne mélyebben is foglalkozni a témával, elegendő lehet az egyiket részletesen bemutatni, a többit pedig csak említés szintjén megjeleníteni. A QR-kódok hibajavító algoritmusáról is kézenfekvő lenne beszélni, ám ennek matematikai háttere már jóval összetettebb, ezért egy átlagos 10. évfolyamos osztályban nem tartjuk reális célnak ennek a teljes megértését.

Bankszámla és bankkártya számok

Luhn algoritmus bankkártyaszámmal:



1. A jobb szélső számjegytől kezdve minden második számjegyet megduplázzuk.
2. Az így kapott számok számjegyeit összeadjuk (a duplázás után kapott kétjegyű számok így két számjeggyel járulnak hozzá az összeghez).
3. Az ellenőrzőszám az előbbi összeget 10-zel oszthatóra egészíti ki.

Az algoritmus célja: elütések és egyszerű számfelcserélések felismerése bankkártyaszámokban és más azonosítókban.

Feladat lehet például, hogy egy régi bankkártyán határozzák meg a tanulók az ellenőrzőszámot, vagy az ellenőrzőszám segítségével valamelyik hiányzó számjegyet.

Néhány lehetséges kapcsolódó kérdés:

Kiszűri a felcserélt számjegyeket az algoritmus? Vissza tudunk-e következtetni arra, hogy melyik két szám cserélődött fel?

Megoldás:

Csak speciális esetekben szűri ki a felcserélt számjegyeket. A leggyakoribb hibát, az egymás melletti számok cseréjét viszont kiszűri, hiszen ekkor más számot fogunk duplán venni az összegben.

💡 Megj.: Ha páros és páratlanadik számjegyet cserélünk fel, ezeket kiszűri az algoritmus.

Visszakövetkeztetni, hogy mely számok cserélődtek fel, szintén csak bizonyos esetekben tudunk. Megvizsgálhatjuk, hogy melyik két számjegy felcserélésével tudjuk úgy kompenzálni az összeget, hogy az ellenőrzőszám megfelelő legyen.

- 1-el tudjuk például növelni az összeget ha x és x+1 számok esetén az x helyett az x+1-et duplázzuk meg.
- 2-vel tudjuk növelni az összeget ha x és x+2 számoknál az x helyett az x+2-t duplázzuk meg.
- k-val növelhető az összeg, ha x és x+k számoknál az x helyett az x+k-t duplázzuk.

Fordítva ugyanígy tudjuk csökkenteni.

Tehát azt meg tudjuk mondani, hogy a két felcserélt szám különbsége mennyi, de ez több lehetséges számpárnál is előfordulhat.

Mennyi annak az esélye, hogy egy random generált bankkártyaszám átmegy a Luhn teszten?

Megoldás: mivel a végén a 10-es maradékot nézzük, a valószínűség 1/10.

Személyi szám

A személyi szám 11 számjegyből áll, M ÉÉHHNN SSS K formában.

- M: az első számjegy a nemre és a születési évszázadra utal
- ÉÉHHNN: a születési év utolsó két számjegye, a hónap és a nap
- SSS: azonos napon születettek megkülönböztetésére szolgál.
- K: ellenőrző számjegy, amelyet a többi 10 számjegyből számolnak ki, hogy az elütések, egyszerű hibák felismerhetők legyenek.

Ellenőrzőszám képzése:

- Régi szabály (1996.12.31-ig születetteknél):

Az ellenőrzőszám az $1 \cdot k_1 + 2 \cdot k_2 + ... + 9 \cdot k_9 + 10 \cdot k_{10}$ összeg 11-es maradéka, ahol $k_1, k_2 ... k_{10}$ a személyi szám első 10 számjegyét jelöli.

- Új szabály (1997.01.01-től):
  Az ellenőrzőszám a $10 \cdot k_1 + 9 \cdot k_2 + ... + 2 \cdot k_9 + 1 \cdot k_{10}$ összeg 11-es maradéka, ahol $k_1, k_2 ... k_{10}$ a személyi szám első 10 számjegyét jelöli.

Ha az eredmény 10 lenne, ilyen számot nem adnak ki. Az ellenőrzés segít kiszűrni a legtöbb elgépelést vagy számfelcserélést.

A következő játékban az ellenőrzőszám segítségével visszakövetkeztethetünk a hiányzó számjegyre: https://vizilo11.github.io/szemelyiszam/

Lehetséges kapcsolódó kérdés:

Miben jobb az új szabály? Milyen hibát szűr ki amit az előző nem szűrt ki?

Megoldás: A régi szabály nem szűri ki azt a hibát, amikor az ellenőrzőszámot és az előtte lévő számot cseréljük fel.

Taj szám

Ehhez érdekes lehet a következő, 2025. októberi emelt érettségi 9. feladata: https://dload-oktatas.educatio.hu/erettsegi/feladatok_2025osz_emelt/e_mat_25okt_fl.pdf

A TAJ szám (társadalombiztosítási azonosító jel) egy 9 karakterből álló azonosító kód, amelyben az első 8 karakter egy-egy számjegy, a kilencedik karakter pedig az első 8 számjegyből képzett ellenőrző számjegy. Az ellenőrző számjegy képzési szabálya: az első nyolc számjegy közül (elölről nézve) a páratlan sorszámú helyeken álló számjegyeket 3-mal, a páros sorszámú helyeken állókat pedig 7-tel szorozzuk. E szorzatok összegének utolsó számjegye lesz az ellenőrző számjegy.

a) Határozd meg a TAJ szám ellenőrző számjegyét, ha az első nyolc számjegy 24165379.

b) Egy TAJ szám első számjegyét letakartuk, így az _14564797 számsor látható. Határozd meg a letakart számjegyet!

c) Határozd meg az ellenőrző számjegy lehetséges értékeit abban a TAJ számban, amely 02563abba alakú! (a és b nem szükségképpen különböző számjegyek.)

💡 Megj.: Többek között a könyvek ISBN kódjai és a bankszámlaszámok is hasonló ellenőrzőszámos algoritmust használnak.

Bináris keresés

A számrendszerek egyik további alkalmazása a bináris keresés, amely során a kettes számrendszer logikáját felhasználva kereshetünk meg egy adott elemet egy listában vagy adatbázisban. Ezt a módszert például a helyesírás-ellenőrző programok is alkalmazzák: a szöveg szavait összehasonlítják egy adatbázisban tárolt szókészlettel, és a megfelelő elemet bináris kereséssel azonosítják.

💡 Megj.: Hasonló feladat a telefonkönyv egyik nevének kitalálása is.

A bináris keresés lényege, hogy minden lépésben felére csökkenti a szóba jöhető lehetőségek számát. Gondolhatunk erre úgy, hogy az adatbázis elemei sorba vannak rendezve, és minden keresésnél a középső elemhez viszonyítunk, így minden lépéssel felezzük a keresési tartományt. Ugyanez az alapelv érvényesül akkor is, ha az adatbázis elemeihez kettes számrendszerbeli számokat rendelünk, és a keresés során ezek számjegyeire kérdezünk rá egyesével.

Ehhez kapcsolódóan az alábbi két játékos feladatot javasoljuk. Ezek viszonylag rövid időt igényelnek, ezért akár a témától függetlenül is beilleszthetők óra eleji ráhangolódásként.

Keresés rendezett listában

A listában megtalálod az első 20 leggyorsabb állatot, csúcssebességük szerint.

1. Vándorsólyom (389 km/h)
2. Aranysas (321 km/h)
3. Bögöly (145 km/h)
4. Fekete márna (129 km/h)
5. Vitorláshal (109 km/h)
6. Gepárd (120 km/h)
7. Anna-kolibri (98 km/h)
8. Villásszarvú antilop (89 km/h)
9. Ugrálóantilop (88 km/h)
10. Gnú (82 km/h)
11. Oroszlán (81 km/h)
12. Feketeantilop (80 km/h)
13. Mezei nyúl (73 km/h)
14. Afrikai vadkutya (72 km/h)
15. Strucc (71 km/h)
16. Kenguru (70 km/h)
17. Agár (69 km/h)
18. Prérifarkas (68 km/h)
19. Zebra (64 km/h)
20. Zsiráf (60 km/h)

Feladat: Minél kevesebb barchoba kérdéssel helyezd el a képen látható Szivárvány villámot a rangsorban.

💡 Megj.: a barchoba kérdés olyan kérdésre utal, aminek csak az igen vagy a nem a két lehetséges kimenetele.

💡 Megj.: Szivárvány villám egy AI által generált, nem létező állat. Jó hír, hogy szabadon megválaszthatjuk, hogy éppen mennyi legyen a csúcssebessége. :)

Megoldás: Ha ügyesen kérdeznek a diákok, 5 kérdés elegendő. A lényeg, hogy olyan kérdéseket tegyenek fel, amikkel mindig két egyenlő (vagy páratlan osztása esetén 1 különbségű) halmazra osztják fel a listát. Célszerű azzal kezdeni, hogy gyorsabb/lassabb-e, mint a 10. állat. Ezt követően mindig a lehetséges intervallum felére kell kérdezni. Persze kissé kacifántosabban más kérdésekkel is elérhetjük, hogy mindig azonos (vagy közel azonos) méretű halmazra osszuk fel a szóba jövő lehetőségeket.

Tanárként célszerű, ha mindig azt a választ adjuk amivel a lehető legtöbb állat marad még versenyben. A végén fel lehet fedni a titkot, hogy Szivárvány villámnak igazából nincsen helye a listában. Az ő kérdéseik függvényében pakoltuk valahová, mindig gonosz módon úgy, hogy minél később tudják csak kitalálni a helyét.

További kapcsolódó kérdés:

10 kérdés mekkora listához elég?

Megoldás: Az előző logikát követve rájöhetnek a tanulók, hogy 1 barchoba kérdéssel a legjobb esetben is felezni tudják a szóba jövő lehetőségeket. Így 1 kérdés legfeljebb két elemre, 2 kérdés 4 elemre ... 10 kérdés legfeljebb $2^{10} = 1024$ elemre elég.

Keresés nem listaszerű elemek között

Az előzőnek folytatása lehet az a típusú barchoba, amikor nem egy listából hanem elszórt elemekből kell kitalálni, hogy melyikre gondoltunk.

Melyik madárra gondoltam az alábbiak közül?

Azt feltételezzük, hogy eleinte a diákok a madarak tulajdonságaira próbálnak rákérdezni és ez alapján szűkíteni a lehetőségek számát. Egy idő után azonban bízunk benne, hogy észreveszik, hogy ha a madarakhoz számokat rendelnek, akkor a korábbi, listánál használt módszer működik. Ebből levonhatjuk azt a következtetést, hogy ez a módszer nem csak rendezett listáknál működik, hanem ennél sokkal tágabb környezetben is.