

A program a következő játékot valósítja meg.¹ Autók rohannak a képernyőn alulról felfelé, több sorban, párhuzamosan, azonos sebességgel. A mi autónkat egy *csillag* (szorzójel) képviseli. Ezt a csillagot mozgatjuk a képernyő 7. sorában vízszintesen jobbra (az *L* gombbal) vagy balra (az *A* gombbal). Ha a képernyő bal oldaláról átérünk a csillaggal a jobb oldalra anélkül, hogy a felfelé áramló autókba ütköznénk, akkor nyertünk. A gép közben számolja az eltelt időt. Ha karambolozunk, akkor a gép *ezt* írja ki, továbbá azt, hogy milyen messze jutottunk el a képernyő bal szélétől.

Megjegyzés. Mivel az autók gyorsan rohannak felfelé, az adott magasságban mozgó csillag azt a benyomást kelti, mintha az autósorral szemben haladna.

A játék sebességéről. A program egyszerűbb változatában egy autót egy (vagy három) nulla jel képvisel. Ha minden pillanatban külön nyomtatnánk az autókat oda, ahol lenniük kell, az élvezhetetlenül lassúvá tenné a játékot. Ehelyett a legalsó sorba nyomtatjuk az autókat, amiktől az *előzőleg kinyomtatott autók automatikusan egy sorral feljebb csúsznak*. Ez a soremelés adja az autók felfelé áramlását, ez az egyik oka, hogy programunk kielégítően gyors. Még mindig nagyon lassú lenne azonban, ha a legalsó sorba a hat-nyolc autót külön-külön nyomtatnánk. Ezért egy olyan A(I)$ string-sorozatot képezzünk, amely betűközökből és az autóknak megfelelő alakzatokból áll, azt nyomtatjuk $I = \emptyset, 1, \dots, 9$ -re a képernyő aljára, majd újra kezdjük $I = \emptyset$ -nál. A gyorsítást tehát az adja, hogy például az

$$A$(I) = " \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset "$$

kinyomtatása egyetlen lépés, de 9 autó megjelenítése (feltéve, hogy egy autót éppen egy \emptyset -val ábrázolunk).

A csillag árnyéka. A programban a csillag a felülről számított 7. sorban mozog jobbra-balra. Egy-egy autósor kinyomtatásakor a csillag is feljebb csúszik, majd egy újabb csillagot nyomtatunk ki a 7. sorba. Így azután általában több csillagot is látunk a képernyőn, de csak egyik „érvényes”. Ez ellen lehetne védekezni az „árnyékcillagok” kiirtásával. Alább találsz erre is egy feladatot.

Egy feladatsorral kezdjük: ha ezeket a feladatokat megoldottad, már könnyű lesz megírnod, vagy legalább is megértened az Őrült Autós programot.

Az alábbiakban hosszú stringeket készítünk. Ilyenkor gyakran nem elég az az 50 hely, amit a gép a string-műveletekre *automatikusan* biztosít. Ezt OS hibaüzenet jelzi. Biztosíthatjuk magunkat ez ellen, ha a program elejére elhelyezünk egy CLEAR 500 vagy CLEAR 1000 utasítást. A gép CLEAR N hatására N byte-ot (helyet) köt le string-készítésre (string-műveletekre). (A DIM-eket tegyük a CLEAR után!)

A játékot két változatban fogjuk megépíteni. Az első változatban három egymás alatti \emptyset jelöl egy autót, a másik változatban megtanulunk bonyolultabb „grafikus” alakzatokat nyomtatni, és ilyenekből állítjuk elő az autókat.

Az első változatban az autókat „egyesével” írjuk be egy sémába: az A(I)$ -ket magunk készítjük el. A második változatban megadjuk a gépnek, hogy milyen sűrűn akarjuk az autókat. Ebből az A(I)$ -ket a gép állítja elő.

Az első változatban nem foglalkozunk az adminisztrációval, mindent a legegyszerűbb formában intézünk el. A második változat már majdnem kidolgozott változat: jól működik és legfeljebb apróságokat érdemes javítani rajta.

Mindkét program *azonos felépítésű*: egybevezető résszel kezdődik, majd egy ciklus következik, amelyben

- kinyomtatunk egy új autósor,
- mozgatjuk az Őrült autóját,
- ellenőrizzük, nem karamboloztunk-e.

A befejező rész jelzi, hogy nyertünk-e vagy karamboloztunk (és a jobb változatban kijelzi a teljesítményünket is).

Feladatok

1. Készíts egy ciklust, amelyik az alábbi A(I)$ -t nyomtatja ki ciklikusan $I = \emptyset, 1, 2, \dots, 9$ -re, majd előlről $I = \emptyset, 1, 2, \dots, 9$ -re:

1985-02-070-1.eps

(Az mindegy, hogy hogyan töltöd ki a helyeket nullákkal és betűközökkel, de végül ezek sűrűsége és szabálytalansága fogja meghatározni a játék nehézségét.)

2. Módosítsd az előző feladat programját úgy, hogy a gép mindegyik A(I)$ -t háromszor nyomtassa ki egymás után. Lásd el a programot egy INPUT V kezdéssel és egy $100/V$ -es lassító ciklussal²; V lesz az autók áramlásának sebessége.

3. Készíts egy programot, amelyik a fentiek mellett egy csillagot (szorzó-jel) mozgat a 7. sorban jobbra-balra (INKEY\$-ral, A és L hatására).

4. Írj egy programot, amelyik bekér 10 egész számot 0 és 1023 között és PRINT @ K , "X" hatására X -eket tesz a megfelelő helyekre. Ezután INKEY\$-ral mozgathatsz egy csillagot a képernyőn négy irányba és ha olyan helyre érsz, ahol X áll, akkor a gép kiírja: „Karamboloztál!”

5. Próbáld ki a következő programot $J = 1, 2, 3$ -ra:

¹ Részlet a szerző „Számítástechnika” c. tankönyvéből.

² Ez a következő üres magú ciklust jelenti: FOR $I=1$ TO $100/I$: NEXT I .

Azt tapasztaljuk, hogy egy betű helyére különböző geometriai alakzatokat is nyomtathatunk.

Érdekesebb alakú autókat (UFO-kat) gyárthatunk, ha a következő programsort használjuk:

Mi a CHR\$(190)+CHR\$(189) alakú autókat fogjuk használni.

6. A fenti feladatok alapján írd meg az Őrült Autós program egyszerűbb változatát az alábbi segítséget is figyelembe véve:

- (i) A programot kezd a

sorral. Az ezután következő sorok definiálják A\$(I)-t $I = 1, 2, \dots, 10$ -re az 1. feladatban leírt módon. (Amikor már működik a program, akkor változtathatsz a 0-k sűrűségén, hogy a játékot élvezetessé tedd.)

- (ii) Mindegyik A\$(I)-t háromszor nyomtasd ki, mielőtt a következőre áttérnél.
- (iii) A csillagot a

PRINT @384 + K, " * "; PRINT @960, ""

segítségével nyomtasd ki. Itt a második PRINT miatt fog a következő PRINT A\$(I) a legalsó sorba nyomtatni. (Egy másik megoldás: PRINT @960, A\$(I)).

- (iv) Akkor nyersz, ha az INKEY\$ által szabályozott K értéke eléri 55-öt.

(v) Amikor a KARAMBOL-t ellenőrzöd, akkor nem arra vagy kíváncsi, hogy A\$(I) K-adik helyén 0 van-e vagy sem, hiszen a kocsi magasságában ilyenkor éppen A\$(I - 2) van, vagy valami ilyesmi: egy korábbi A\$(H). Gondold át, hogyan tudod H-t úgy megkapni, hogy az

IF MID\$(A\$(H), K, 1) = " 0"

feltétel éppen akkor teljesüljön, ha a csillagod egy 0-t talál el. (Attól függően, hogy hogyan valósítod meg a programot, néha csak minden második lépésben nyomtat ki A\$(I)-t, minden második sort üresen hagy, vagy esetleg mindegyik sorba nyomtat. Mindenesetre három autósorral lejjebb A\$(H + 1), hat sorral lejjebb A\$(H + 2) van, ...)

- 7. Írd meg az Őrült Autós játék fejlettebb változatát is, az alábbiak alapján:

A *bemenő adatok*: kocsiáradat sebessége V, kocsik sűrűsége $P \in (0, 1)$, a véletlen szám generátor „magja” Q.

- (i) Vedd alapul az egyszerűbb változatban elmondottakat.

(ii) Készíts egy véletlenszám-generátort, amely 100 (0, 1) közötti számot ad ki. X_1, X_2, \dots, X_{100} -at. Egy INPUT-tal bevitt P-re a gép ellenőrizze, $X_1 < P$ teljesül-e. Ha igen, akkor (definíció szerint) legyen $A(I) = 1$, egyébként $A(I) = 0$!

(iii) Készíts 10 darab A\$(I) vektort, melyek mindegyikében legyen 10 darab 2 szélességű hely, ahová autókat tehetünk és bármely két ilyen hely között két kihagyás:

Ahová XX-et raktunk, oda tegyél CHR\$(190) + CHR\$(189)-et vagy hagyj ott két üres helyet *attól függően*, hogy $A(10 * I + J) = 1$ vagy 0. Így az autók sűrűsége hozzávetőlegesen P lesz: $P = 0.3$ esetén 30 körül fog járni.

(iv) Ezekkel az autókkal készítsd el az autók felfelé áramló seregét. Az űrült autós autója most "##" legyen, tehát szintén kettő szélességű. Így jobban is látszik.

- (v) Számold az eltelt időt. A végén nyomtasd ki.

(vi) A programot kezd egy megfelelő bevezető résszel, melyben ismerteted, hogyan kell használni a játékot.

(vii) Ha karamboloztál vagy ha nyertél, 10 másodperc után a gép kezdjen új játékot az eredeti pályán. (Ha új pályára vágysz vagy új játéksűrűségre, BREAK + RUN-nal indíts újra.)

Megjegyzés. A fentiekben A\$(I) végére 12 szabad helyet raktunk, hogy A\$(I) 54 hosszúságú legyen. Lehetne még hosszabb is, de ha rövidebbre választjuk, hibajelzést kapunk karambol-ellenőrzéseink során. (Miért?)

Rátérünk a programra. Rögtön a fejlettebb változatot adjuk meg, a 7. feladat megoldását.

Magyarázat

A program 10 sorában levő CLEAR 1000 biztosítja a helyet a gép memóriájában az „AUTÓK KÉSZÍTÉSE” egységben történő string-műveletekhez. Ha kihagynánk, OS hibaüzenet jelezné, hogy e string-műveletekre szánt 50 hely nem elegendő. A 30–120 sorok közlik a játékra vonatkozó információkat. 210–230-ban egy 10×10 -es véletlen mátrixot (100 hosszúságú A(N) vektort) készítünk a 100-ban bekért Q négyzetgyökével (amit szintén Q-val jelölünk)

és a 110-be bekért $0 < P < 1$ számmal. Minél nagyobb P , annál sűrűbbek lesznek 1-ek $A(N)$ -ben és annál több autót fogunk generálni az „AUTÓK KÉSZÍTÉSE” elnevezésű egységben. A játék élvezhető már $P = 0.3$ -mal is, nehezzé kezd válni $P = 0.45$ -nél és lényegében megnyerhetetlen $P = 0.55$ -nél. A 250 – 260 sorok kinyomtatják a kapott $A(N)$ -eket 10×10 -es elrendezésben (mátrixban). Ott lesznek az autók, ahol most 1-eket látunk. Ha ezt az ellenőrzést feleslegesnek tadd REM-be.

Az adatok bevitele automatikus értékadással történik: valahányszor az INPUT ... megkérdez egy adatot Q , T és P közül, válaszolhatunk a szabályoknak megfelelően, de az is elég, ha lenyomjuk NL-t: előzőleg ugyanis már választott a program értéket ezen adatok számára, így azonnali NL esetén megtartja ezeket: $Q = 2$ majd $\sqrt{2}$, $T = 10$, $P = 0.4$.

Az AUTÓK KÉSZÍTÉSE egységnél vigyázzunk a megfelelő számú betűközre! A felfelé futó autók mindig két karakter szélességűek, és a 4., 8., 12., 16., ... oszlopokban kezdődhetnek. Ez azt jelenti, hogy két autóoszlop között mindig lesz két egységnyi *pihenőhely* a mi két karakter szélességű autónknak is. Ez megkönnyíti a játékot. A 330 sor aktiválása (REM elhagyása) esetén a sorok fele beljebb kezdődik, megszűnik a pihenő-sáv, megnehezedik a játék.

1985-02-073-1.eps

A 320 – 370 sorokban $B\$$ éppen $A\$(I)$ már elkészült része. Mindenütt írhattunk volna $A\$(I)$ -t is $B\$$ helyett. Próbáld ki, mi történik, ha $A\$(I)$ végére nem teszel (elég) üres helyet (370-es sor). K számolja, mennyit haladunk jobbra, FI pedig a futási időt.

430 hatására a gép kinyomtat egy üres sort a képernyő aljára, ezután kinyomtatja a soron következő autókat ugyanide, de persze előbb megemel mindent egy sorral. Ez tulajdonképpen az egész autóáramlás alapja. Ha csak PRINT @960, $A\$(I)$ -t használnánk 430 helyett, az autók áramlata kétszer ilyen sűrűvé válna. (Ilyenkor a karambol ellenőrzés megváltozik!)

A 440 sor intézi el, hogy $A\$(0)$, ... $A\$(9)$ után megint $A\$(0)$ következzen kinyomtatásra.

460 folyamatosan mutatja az eltelt időt a bal felső sarokban. Ha úgy találsz, hogy idegesítően gyorsan fut, FI helyett $INT(FI/4)$ -et nyomtasd ki.

A 450–500 ciklus várja, hogy lenyomjunk egy gombot: A -t vagy L -et, majd egy idő után „lejár”: 510 kinyomtatja a képernyő felső harmadába a mi „autónkat”, $###$ -ot. Ha viszont lenyomjuk A -t vagy L -et, a gép nem várja meg a Z -ciklus lejártát. Így a játék akkor is felgyorsul, ha lassúra állítjuk be, de gyorsan nyomogatjuk A -t és L -et. Ez izgalmasabbá teszi a játékot.

A KARAMBOL? egységben ellenőrizzük, hogy nem került-e ugyanoda a $###$ és egy $A\$(J)$ autója. Ha nem, akkor 550 visszaküld 430-ra: kinyomtat egy új autósort és várja, hogy lépünk-e. Ha viszont karamboloztunk, 560 jelzi ezt, K -t is kinyomtatva. Így megtudjuk, hogy milyen messzire sikerült elkerülnünk a balszéltől.

Ha $###$ -gal elérkeztünk az 54 pozícióra, akkor NYERTÜNK, amit a program 520 sora jelez ki.

Mindkét esetben az 570 sornál kötünk ki, ahol egy HAMIS INPUT állítja le a gépet. Leolvashatjuk az eltelt időt, és eldönthetjük, hogy NL-lel folytatjuk-e a játékot a megelőző pályán és sebességgel, vagy BREAK + RUN-nal új játékba kezdünk.

Feladatok

1. Tüntesd el a játékból az árnyék-csillagokat!
2. Módosítsd a játékot úgy, hogy ne függjön az autók áramlási sebessége attól, hogy milyen gyorsan nyomogatjuk A -t és L -et!
3. Módosítsd az autók generálását úgy, hogy különböző alakú autók is megjelenjenek a pályán!