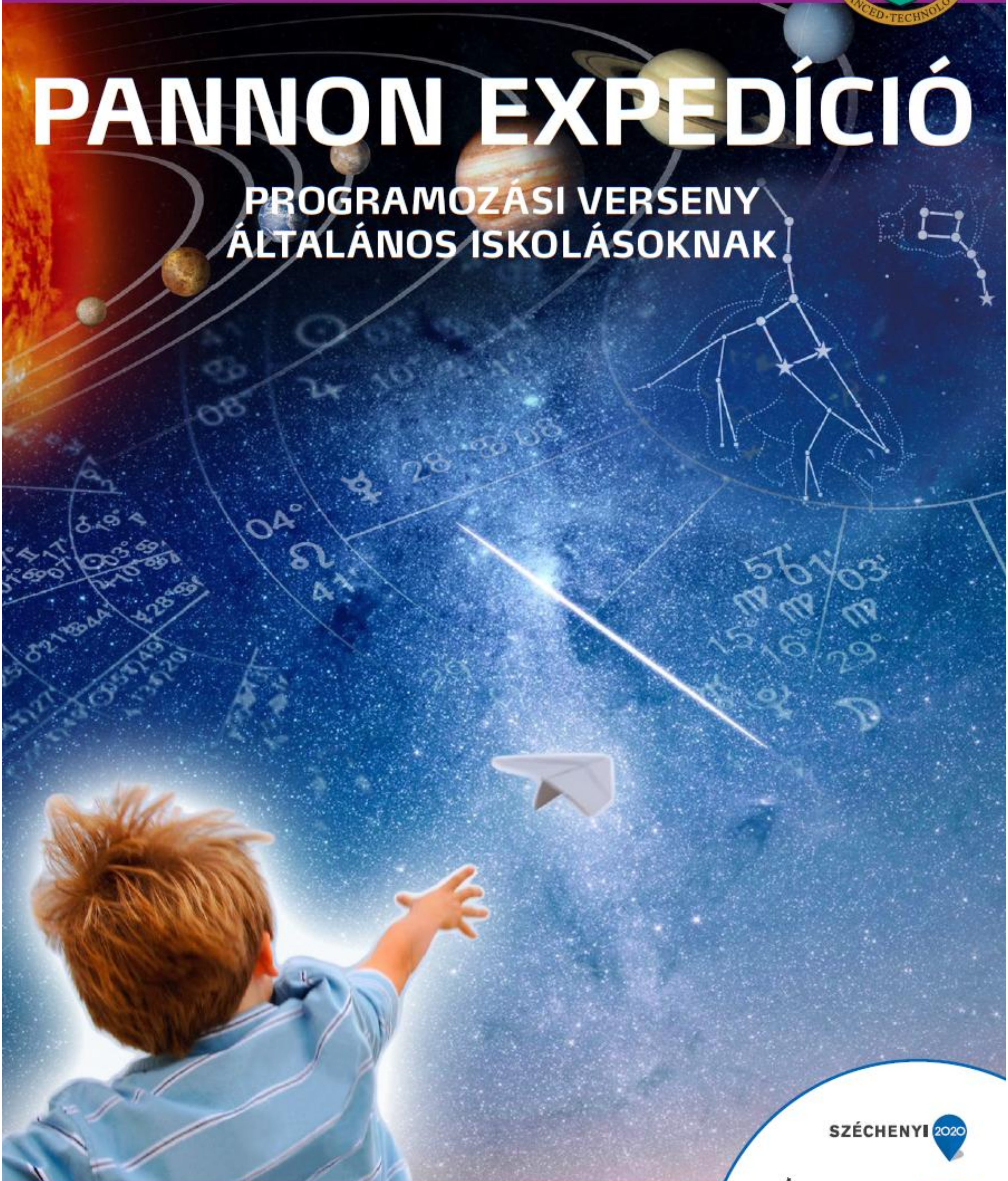
PANNON EGYETEM
MŰSZAKI INFORMATIKAI KAR

PANNON EXPEDÍCIÓ

PROGRAMOZÁSI VERSENY
ÁLTALÁNOS ISKOLÁSOKNAK



https://pexpedicio.mik.uni-pannon.hu

SZÉCHENYI 2020
Magyarország Kormánya
Európai Unió
Európai Szociális Alap
BEFEKTETÉS A JÖVŐBE

A kiadvány, a rendezvény és a verseny az **EFOP-3.4.4-16-2017-00002** azonosító-számmal ellátott

"*A felsőoktatásba való bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a Pannon Egyetemen*”

elnevezésű projekt keretében valósulhatott meg.

Jelen kiadvány elérése: https://pexpedicio.mik.uni-pannon.hu/images/PExFuzet.pdf

A füzetet A/5 méretű (kicsinyített) nyomtatásra terveztük.

A **Pannon Expedíció**

(PExpedíció)

verseny

2019.

Feladatok és megoldások

dr. Szalkai István
szalkai@almos.uni-pannon.hu

2019. április 27.

Bevezetés

Üdvözlünk

a csillagközi Pannon Expedíció (PExpedíció) irányítóközpontjában! El kell juttatnunk egy űrhajót a 100 évvel ezelőtt (2020-ban) elindult telepesek utódainak felkeresésére!
Ráadásul a 8 fordulós verseny végén értékes nyereményekkel gazdagodhatnak a résztvevők!

A legelső feladat 2019. január 1-jén kedden, 18:00:00 órakor, a többi feladat kéthetente, minden második hétfőn 18:00:00 órakor jelenik meg a verseny honlapján:
https://pexpedicio.mik.uni-pannon.hu/
Egészen az utolsó forduló megkezdéséig lehetőséged van a versenybe bekapcsolódnod!

A harmadik és a hatodik feladat után lehetőséged lesz a feladatok próba feltöltésére: visszajelzést kapsz megoldásaid helyességéről. Minden segédeszközt (felnőtteket is) szabadon használhatsz!

A 8. feladat megjelenése után, április 08., 18:00:00 -tól mind a nyolc feladat helyes megoldását ismét fel kell töltened, és a leggyorsabbak kapnak meghívást a Döntőbe, ahol már csak egy-két feladatot kell önállóan megoldanod, és már Tied is lehet a Fődíj!

Segítségedre lehetnek a tavalyi PEnigma verseny feladatai és megoldásai: PEnigma füzet: https://pexpedicio.mik.uni-pannon.hu/images/PEnigmaFuzet.pdf

A feladatokat **Süle Péter**nek és **Szalkai István**nak köszönhetjük!

Veszprém, 2019. április 27.

Pannon Egyetem, **M**űszaki **I**nformatikai **K**ar, Matematika Tanszék

PANNON EGYETEM
MŰSZAKI INFORMATIKAI KAR

1. FELADAT: PRÓBAREPÜLÉS

Legelőször is be kell gyakorolnunk az űrhajó irányítását, térbeli helyzetének kiszámítását.

Ülj le egy nagy (torna- vagy képzeletbeli-) terem sarkába úgy, hogy jobb lábad a jobboldali-, míg bal lábad a bal oldali fal mellett legyen vízszintesen a földön, törzsed függőlegesen felfelé álljon a két fal találkozásánál. Vegyél öledbe (vagyis pontosan a terem sarkába) egy papírrepülőt, amelynek orra nézzen a jobb lábfejed irányába vízszintesen (a parkettán), és irányítsd az alábbi parancsok szerint:

Előre (E) = előre 1 dm, amerre éppen a repülő orra áll,
Jobbra (J) = a gép jobbra fordul 90° -ot (a saját orra iránya szerinti jobbra!),
Balra (B) = a gép balra fordul 90° -ot (a saját orra iránya szerinti balra!),
Felfelé (F) = a gép "felfelé" fordul 90° -ot (a saját orra iránya szerinti "felfelé"!),
Lefelé (L) = a gép "lefelé" fordul 90° -ot (a saját orra iránya szerinti "lefelé"!),

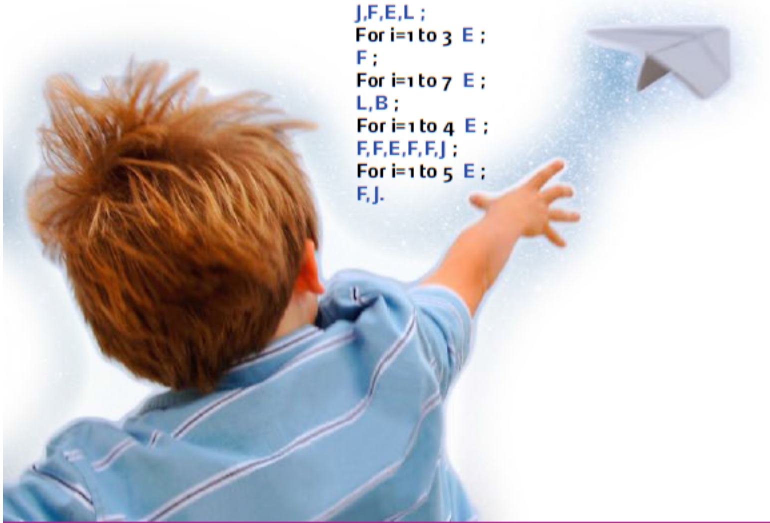
A J,B,F,L "forduló" parancsoknál a gép térbeli helyzete nem változik, csak iránya, az E parancsnál halad a gép, de iránya nem változik.
Vigyázz: Például két "felfelé" (F,F) parancs után a gép (és benne a pilóta) fejjel lefelé repül visszafelé irányban!, és ekkor a gépet felülről nézve a mi jobb- és bal kezünk éppen ellentétes a pilóta jobb és bal kezével, márpedig a J és B parancsok mindig a pilóta kezeinek irányára vonatkoznak!

Másik példa: ha a gép vízszintesen (szokásos módon) repült, akkor az F,J vagy F,B parancsok után ismét vízszintes irányban fog repülni, de szárnyai (szárnysíkjai) függőlegesen fognak állni!
Tehát mindig képzeld bele magad a gépben ülő pilóta helyzetébe!

Indulás: A legelső E parancs hatására a gép vízszintesen indul a jobboldali lábad irányába a fal mellett a parkettán, felemelkedni csak a legelső F,E parancsok után fog 1 dm -re a parkettától, és ezek a parancsok után a gép (és a pilóta) orra felfelé áll. A következő L parancs után fog ismét "normálisan" repülni.

Program:

```
For i=1 to 2 E ;
B ;
For i=1 to 6 E ;
J,F,E,L ;
For i=1 to 3 E ;
F ;
For i=1 to 7 E ;
L,B ;
For i=1 to 4 E ;
F,F,E,F,F,J ;
For i=1 to 5 E ;
F,J.
```



A kérdés: Hol van a program végrehajtása után a repülőgép?

Beküldendő a hat számjegyből álló kód: bbjjcc, ahol bb és jj a repülőgépnek a bal- illetve a jobboldali faltól való távolsága, cc pedig a parketta feletti magassága dm-ben. Amennyiben valamelyik érték 10-nél kisebb, akkor az első számjegye 0.

Segítség:
A program minden ciklusa és sora után írjuk fel, hogy a gép éppen hol tartózkodik: [x,y,z] (dm), orra merre áll: (u,v,w), és a pilóta "lába->feje" vektor merre áll: <a,b,c>.
Mindegyik esetben a három koordináta sorrendje: jobb lábam irányában, bal lábam irányában, felfelé.

SZÉCHENYI 2020

https://pexpedicio.mik.uni-pannon.hu

Magyarország Kormánya
Európai Unió Európai Szociális Alap
BEFEKTETÉS A JÖVŐBE

Az 1. feladat megoldási ötlete

A program minden ciklusa és sora után írjuk fel, hogy a gép éppen **hol** tartózkodik: **`[x,y,z]`** (dm), **orra** merre áll: **`(u,v,w)`**, és a pilóta "lába->**feje**" vektor merre áll: **<a,b,c>**.
Mindegyik esetben a három koordináta sorrendje: jobb lábam irányában, bal lábam irányában, felfelé.

		HOL	MERRE	FEJE .
Indulás:		[0, 0, 0],	(1, 0, 0),	< 0, 0, 1> ,
Program:				
For i=1 to 2	**E**;	[, ,],	(, ,),	< , , > ,
	B;	[, ,],	(, ,),	< , , > ,
For i=1 to 6	**E**;	[, ,],	(, ,),	< , , > ,
	J,	[, ,],	(, ,),	< , , > ,
	F,	[, ,],	(, ,),	< , , > ,
	E,	[, ,],	(, ,),	< , , > ,
	L;	[, ,],	(, ,),	< , , > ,
For i=1 to 3	**E**;	[, ,],	(, ,),	< , , > ,
	F;	[, ,],	(, ,),	< , , > ,
For i=1 to 7	**E**;	[, ,],	(, ,),	< , , > ,
	L,	[, ,],	(, ,),	< , , > ,
	B;	[, ,],	(, ,),	< , , > ,
For i=1 to 4	**E**;	[, ,],	(, ,),	< , , > ,
	F,**F**,	[, ,],	(, ,),	< , , > ,
	E,	[, ,],	(, ,),	< , , > ,
	F,**F**,	[, ,],	(, ,),	< , , > ,
	J;	[, ,],	(, ,),	< , , > ,
For i=1 to 5	**E**;	[, ,],	(, ,),	< , , > ,
	F,	[, ,],	(, ,),	< , , > ,
	J.	[, ,],	(, ,),	< , , > .

A kitöltött táblázat és a végeredmény a túloldalon van.

Az 1. feladat megoldása

A program minden ciklusa és sora után írjuk fel, hogy a gép éppen **hol** tartózkodik: **[x,y,z]** (dm), **orra** merre áll: **(u,v,w)**, és a pilóta "lába->**feje**" vektor merre áll: **<a,b,c>**.
Mindegyik esetben a három koordináta sorrendje: jobb lábam irányában, bal lábam irányában, felfelé.
Pirossal jelöltük, ami változott az előző sorhoz képest.

			HOL	MERRE	FEJE .
Indulás:			[0, 0, 0],	(1, 0, 0),	< 0, 0, 1> ,
Program:					
	For i=1 to 2	**E;**	[2, 0, 0],	(1, 0, 0),	< 0, 0, 1> ,
		B;	[2, 0, 0],	(0, 1, 0),	< 0, 0, 1> ,
	For i=1 to 6	**E;**	[2, 6, 0],	(0, 1, 0),	< 0, 0, 1> ,
		J,	[2, 6, 0],	(1, 0, 0),	< 0, 0, 1> ,
		F,	[2, 6, 0],	(0, 0, 1),	<-1, 0, 0> ,
		E,	[2, 6, 1],	(0, 0, 1),	<-1, 0, 0> ,
		L;	[2, 6, 1],	(1, 0, 0),	< 0, 0, 1> ,
	For i=1 to 3	**E;**	[5, 6, 1],	(1, 0, 0),	< 0, 0, 1> ,
		F;	[5, 6, 1],	(0, 0, 1),	<-1, 0, 0> ,
	For i=1 to 7	**E;**	[5, 6, 8],	(0, 0, 1),	<-1, 0, 0> ,
		L,	[5, 6, 8],	(1, 0, 0),	< 0, 0, 1> ,
		B;	[5, 6, 8],	(0, 1, 0),	< 0, 0, 1> ,
	For i=1 to 4	**E;**	[5,10, 8],	(0, 1, 0),	< 0, 0, 1> ,
		F,F,	[5,10, 8],	(0,-1, 0),	< 0, 0,-1> ,
		E,	[5, 9, 8],	(0,-1, 0),	< 0, 0,-1> ,
		F,F,	[5, 9, 8],	(0, 1, 0),	< 0, 0, 1> ,
		J;	[5, 9, 8],	(1, 0, 0),	< 0, 0, 1> ,
	For i=1 to 5	**E;**	[10, 9, 8],	(1, 0, 0),	< 0, 0, 1> ,
		F,	[10, 9, 8],	(0, 0, 1),	<-1, 0, 0> ,
		J.	[10, 9, 8],	(0,-1, 0),	<-1, 0, 0> .

Tehát a végső pozíció: [10, 9, 8], (0,-1, 0), <-1, 0, 0> .

A beküldendő kód: **100908**

2. FELADAT: ÚTBAIGAZÍTÁS

Történetünk szereplői megkezdték több hónapos izgalmakban és megpróbáltatásokban gazdag csillagközi útjukat, hogy megtalálják a másik naprendszerbe költözött ősök bolygóját. Útjuk nem kezdődik zökkenőmentesen, most éppen egy kozmikus viharba keverednek, melynek következtében a mágneses erők hatására megzavarodik az űrhajó kommunikációs és navigációs rendszere, így vakon, gépek segítsége nélkül kell folytatni útjukat. Az űrhajó matematikusainak annyit sikerült kideríteniük, hogy a navigációs rendszeren megjelent üzenet egy a Cézár- illetve Vigenére dekódoláshoz hasonló módszerrel megfejthető. Segítsünk szereplőinknek a navigációs rendszer megjavításában és a pontos útirány meghatározásában. A megfejtés megoldása megadja számukra azt a plusz információt, mely alapján tovább folytathatják útjukat és kutatásukat az elveszett civilizáció után.

A szükséges adatokat Cézár kódolással titkosították, a 35 betűs (teljes ékezetes) magyar ABC-t használták, a 36. karakter a szóköz (_). Kis- és nagybetű között nincs különbség, további írásjelek nincsenek. Azonban mindegyik betűt más távolságra tolták el (Vigenère kódoláshoz hasonlóan): az első betűt 1, a második betűt 2, és így tovább, az n-edik betűt (vagy szóközt) n betűvel tolták el.

Például a „Málna_ízű_fagyi" mondat kódja „Ncoőeeóébhójpít".

Feladat: Fejtsük vissza a kapott kódot és hajtsuk végre a pályakorrekciót:

ÁáÓdkípqiuiÖóüsanggpápvejvoéeűfk_aqa

Beküldendő kód: A visszafejtett magyar nyelvű kód, a szokásos magyar helyesírással (első betű nagy, _ helyett szóközök, végén pont).

SZÉCHENYI 2020
MAGYARORSZÁG KORMÁNYA
Európai Unió
Európai Szociális Alap
BEFEKTETÉS A JÖVŐBE

A 2. feladat megoldása

Mind kódoláskor mind dekódoláskor (a feladat megoldásakor) sok betűnek sokszor kell különböző *eltoltjait* kitalálnunk. A betűk eltolását megkönnyíti, ha két **kartonpapír csíkra** számolóléc ("logarléc") módjára felírjuk az abc-t, ügyelve arra, hogy a betűk közötti távolság mindig ugyanannyi legyen (például mindegyik betű egy-egy négyzetrácsba essen), majd a két csíkot egymás alá téve jobbra-balra eltoljuk a megfelelő betűközzel, és máris leolvashatjuk a betűk eltolt értékeit. Az alábbi ábrán **h=10** betűvel toltuk el az alsó abc -t (az alsó szóköz " _ " a felső **h** alatt van), tehát például a **v** betű 10 -dik eltoltja **d** :

_aábcdeéfghiíjklmnoóöőpqrstuúüűvwxyz_aábcdeéfghiíjklmnoóöőp...
<= _aábcdeéfghiíjklmnoóöőpqrstuúüűvwxyz =>

Ha nincs kéznél papír és olló, akkor bármely szövegszerkesztőbe gépeljük be az abc -t két sorban, de monospaced (**egyenlő közű**) betűtípussal, mint például CouierNew vagy Lucida Console, hiszen eltoláskor lényeges, hogy a betűk szélességei *ugyanakkorák* legyenek! Az alsó sort a Space vagy Del gombokkal tologathatjuk jobbra-balra.

A felső sorba érdemes az abc -t kétszer leírnunk egymás után, ugyanis sokszor előfordul, hogy vagy az eltolás túl sok vagy az eltolni kívánt betű (**v**) az abc végén helyezkedik el, ami miatt a betű eltolt értéke (**d**) "kilóg" a fenti abc alól. Nyilván, ha az eltolás mértéke több, mint 35, mondjuk 83, akkor a 83 -ból kivonjuk néhányszor a 35 -öt, példánkban 83-2*35=13, és a betűket (alsó sor) csak 13 -al toljuk el.

Ha *visszafelé* akarjuk tologatni a betűket, akkor érdemes a felső sor második szóközétől " _ " balra tolni az alsó sort, például az alábbi ábrán 6 betűvel toltuk *visszafelé* az alsó sort, ami szerint például az **ö** betű *visszafelé* eltoltja **k** :

_aábcdeéfghiíjklmnoóöőpqrstuúüűvwxyz_aábcdeéfghiíjklmnoóöőp...
<= _aábcdeéfghiíjklmnoóöőpqrstuúüű v

Tehát a feladat megoldása:

A kód betűit egyesével kell *visszafelé* eltolnunk: az első betűt 1-el, a második betűt 2-vel, stb. A megfejtés:

A Magellán Felhő után fordulj jobbra.

Megjegyzés: A *Nagy Magellán felhő* fényes objektum, de csak a déli égbolton látható, az Aranyhal (Dorado) és a Táblahegy (Mensa) csillagképek határán, távolsága tőlünk nagyjából ötven kiloparszek (163 000 fényév) távolságban, vagyis űrhajósaink "még jó előre" megkapták az útbaigazítást. A *Kis Magellán felhő* halványabb, tőle nyugati irányban 20 foknyira található a Tukán csillagképben, egy 3°-os homályos, világos foltként jelenik meg az éjszakai égen, hasonlóan néz ki, mint a Tejút egy kis darabja. Mindkettő törpegalaxis, amely Tejútrendszerünk körül kering, több százmillió csillagot tartalmaz. Mindkettőt *Ferdinand Magellan* portugál felfedezőről nevezték el, aki megfigyelte őket Föld körüli útja során 1519 -ben, de a Nagy Magellán felhőt már 964 -ben '*Abd Al-Rahman Al Sufi* is megemlítette Állócsillagok könyve című művében. https://hu.wikipedia.org/wiki/Nagy_Magellán-felhő , https://hu.wikipedia.org/wiki/Kis_Magellán-felhő .

Érdemes még elolvasni a PEnigma verseny 1. és 2. feladatához írt megjegyzéseket is (6-11., 39. és 45-49. oldalak): https://pexpedicio.mik.uni-pannon.hu/images/PEnigmaFuzet.pdf

3. FELADAT: TÁVOLSÁG MEGHATÁROZÁSA

Történetünk szereplői a csillagközi utazás megkezdése előtt szeretnék kiszámolni, hogy az előzetes információik alapján mekkora utat kell megtenniük a következő években, hogy megtalálják az eltűnt civilizáció nyomait. Mivel előre nem tudhatják, hogy milyen veszélyekkel és megpróbáltatásokkal kell szembenézniük az utazásuk során, így csak megközelítő érték kiszámítására tesznek kísérletet. A számítógép az adatokat „fordított lengyel jelöléssel” (RPL) tárolja és technikai okok miatt így adja át a parancsnoknak:

3; x; 4; -; *; 5; x; *; 2; +; 3; *; +; 6; 2; x; *; -; 2; *; -; 2; x; *; 1; +; 24; 3; /; 2; x; *; *; +; 2; -8; 6; x; *; 2; /; +; *; -; > .

A biztonság miatt kézzel kell ellenőrizni a számítást! Oldd meg az egyenlőtlenséget, ami az út hosszát fényévben adja meg. Kerekítsd felfelé egy egész számra, számold át billió kilométerre, és az eredményt kerekítsd ismét egész (billió) kilométerre.

Beküldendő kód: **a távolság billió kilométerre kerekített értéke** (a végén a 12 nulla nélkül).

Segítség: Az RPL feldolgozásakor a számítógép az adatokat (pontosvesszőkkel) elválasztva kapja, balról jobbra egyesével olvassa be és dolgozza fel.

Ha a soron következő adat egy szám vagy betű (változó), akkor ezt beteszi a (verem-) memória soron következő rekeszébe, balról jobbra. Ekkor a foglalt memóriarekeszek száma eggyel nő.

Ha a következő adat egy műveleti jel vagy az egyenlőtlenség jele, akkor a gép a két utolsó (foglalt) memóriarekesz tartalmát kiolvassa és törli, a műveleti vagy egyenlőtlenség jelet a két, most kivett tartalom közé teszi, és a kapott jelsorozatot a legelső, még üres rekeszbe teszi vissza (tehát a foglalt memóriarekeszek száma eggyel csökken).

VIGYÁZAT: lehetséges, hogy mielőtt a műveleti jelet a két tartalom közé tesszük, a két tartalmat zárójelekbe kell tennünk, hogy matematikailag helyes kifejezést kapjunk!

Alább a feladat **megoldásának elejét** látjuk (a memóriarekeszeket a | jellel választjuk el):

INPUT (pontosvesszőkkel elválasztva)	MEMÓRIAREKESZEK (jobboldalon az utoljára feltöltött)			
3 ; x ; 4	3	x	4	
-	3	x - 4		
*	3*(x -4)			
5 ; x	3*(x -4)	5	x	
*	3*(x -4)	5*x		
2	3*(x -4)	5*x	2	
+	3*(x -4)	5*x +2		
3	3*(x -4)	5*x +2	3	
*	3*(x -4)	(5*x +2)*3		
+	3*(x -4)+(5*x +2)*3			
6 ; 2 ; x	3*(x -4)+(5*x +2)*3	6	2	x
*	3*(x -4)+(5*x +2)*3	6	2* x	
-	3*(x -4)+(5*x +2)*3	6 - 2*x		
2	3*(x -4)+(5*x +2)*3	6 - 2*x	2	
*	3*(x -4)+(5*x +2)*3	(6 - 2*x)*2		
(folytasd . . .)				

https://pexpedicio.mik.uni-pannon.hu

A 3. feladat megoldásának folytatása

A megoldás folytatása:

input	1. rekesz	2. rekesz	3. rekesz	4.rekesz	5.rekesz	6.r.
-	3*(x -4)+(5*x +2)*3-(6-2*x)*2					
2 ; *x* ;	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2	x			
*	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x				
1	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x	1			
+	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1				
24 ; 3 ;	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1	24	3		
/	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1	24/3			
2 ; *x* ;	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1	24/3	2	x	
*	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1	24/3	2*x		
*	3*(x -4)+(5*x +2)*3-(6-2*x)*2	2*x+1	(24/3)*2*x			
+	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x				
2 ; -8 ; 6 ; *x* ;	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2	-8	6	x
*	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2	-8	6*x	
2	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2	-8	6*x	2
/	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2	-8	6*x/2	
+	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2	-8+6*x/2		
*	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x	2*(-8+6*x/2)			
-	3*(x -4)+(5*x +2)*3-(6-2*x)*2	(2*x+1)+(24/3)*2*x-2*(-8+6*x/2)				

	1. rekesz
>	**3*(x -4)+(5*x +2)*3-(6-2*x)*2 > (2*x+1)+(24/3)*2*x-2*(-8+6*x/2)**

Tehát a megoldandó egyenlőtlenség:

$$3(x\text{-}4) + 3(5x+2) - 2(6\text{-}2^*x) > (2x+1) + (24/3)\cdot 2x - 2(\text{-}8+6\cdot x/2)$$

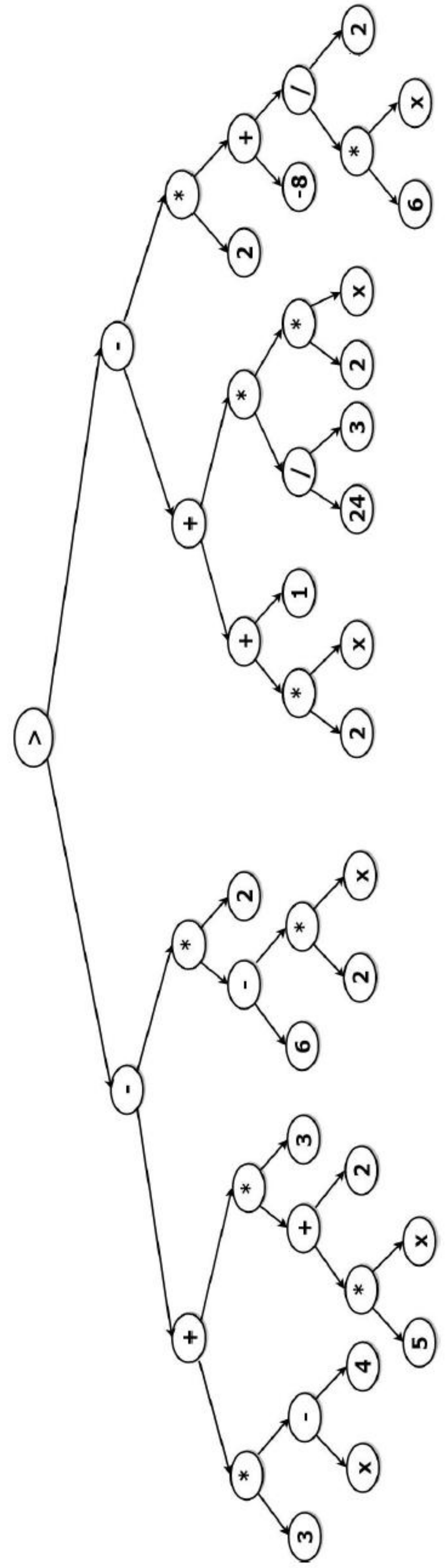
aminek a megoldása: $x > 3{,}5$, vagyis 4 fényév távolságot kell megtenniük az űrhajóval. Mivel 1 fényév = 9,4607 billió km, ezért 4 fényév = 37,8428 billió km, vagyis körülbelül **38 billió km** utat kell megtenniük az űrhajóval.

Megjegyzések: i) Hozzánk a legközelebbi csillag (a Nap után) a **Proxima Centauri** (latinul proxima: legközelebbi) egy vörös törpe a Kentaur (Centauri) csillagképben, a Földtől 4,24 fényév ($\approx 4\cdot 10^{13}$ km) távolságra, 1915-ben fedezték fel.
https://hu.wikipedia.org/wiki/Proxima_Centauri

ii) A feladat szövege elég részletesen leírja a *Fordított Lengyel Logika / Jelölés* (Reversed Polish Logics/Notation, RPL/RPN) megfejtését, azaz jelentését. Ez a számítógépek működésének lényeges része: a műveletek és kifejezések (képletek) tömör tárolása és gyors végrehajtása (kiszámítása) múlik rajta.

Az eredeti (nem fordított) *prefix* jelölési forma a lengyel logika tudósa, **Jan Łukasiewicz** után kapta a nevét, aki 1920-ban kimutatta ennek a módszernek az előnyeit. A tudós származása után módszerét lengyel jelölésnek (Polish notation) kezdték el nevezni. Munkásságát később, 1954- ben Burks, Warren, és Wright, tőlük függetlenül Bauer és Dijkstra az 1960-as évek elején felhasználta a korabeli számítógépek memóriafelhasználásának csökkentésére (akkor még nem voltak gigabájtok). Az eredeti lengyel jelöléssel szemben a fordított (*postfix*) jelölést az ausztrál Hamblin vezette be, amit *fordított lengyel jelölés*nek nevezett el az 1950-es évek közepén, elismerve Łukasiewicz munkásságát. A látszólag aprócska változtatásnak (pre- vagy postfix) óriási jelentősége van a számítógépek működése szempontjából.
https://hu.wikipedia.org/wiki/Fordított_lengyel_jelölés
https://hu.wikipedia.org/wiki/Jan_Lukasiewicz

iii) Az előző oldalon ismertetett megoldás ugyan érthető, de fáradságos. Az RPN kifejezést azonban *hátulról visszafelé olvasva* sokkal könnyebben megérthetjük az eredeti kifejezés szerkezetét és felírhatjuk szokásos formában. Ezt láthatjuk a következő oldalon.



4. FELADAT: A GOMB

A vészjelző automatika működésbe hozta a piros „Memóriakeverés" feliratú gombot! Szerencsére a számítógép memóriájának minden tartalma megőrződött, csak máshová került. Mivel ezt a számítógép részletesen kiírta a monitorra, a másodkapitány azonnal felírta egy papírra:

**„ Az 1.memória tartalma a 2.-ba került,
a 2.memória tartalma az 5.-be került,
a 3.memória tartalma az 1.-be került,
a 4.memória tartalma a 3.-ba került,
... "**

az alábbi táblázatba rendezve:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 2 & 5 & 1 & 3 & 10 & 11 & 7 & 12 & 14 & 4 & 16 & 9 & 17 & 18 & 8 & 19 & 13 & 20 & 6 & 15 \end{pmatrix}$$

A másodkapitány tudja, hogy (ha semelyik másik gombot nem nyomja meg) a „Memóriakeverés" gomb mindig ugyanúgy rendezgeti a memóriák tartalmait, akárhányszor is megnyomhatja. Tehát, mielőtt az űrhajó eltévedne, vissza tudja állítani a számítógép memóriáját az eredeti állapotba!

Még hányszor kell a gombot megnyomni ahhoz, hogy a memória mindegyik rekeszében az eredeti tartalom legyen?

Beküldendő: a legkisebb megfelelő egész szám!

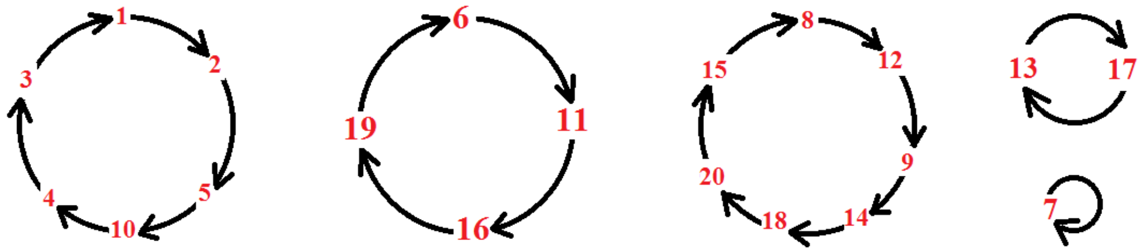
(Vigyázat: a gombot az automatika egyszer már megnyomta, ezt már nem kell számolni!)

A 4. feladat megoldása

Természetesen egy nagy papírra felrajzoljuk a 20 memória (helyét), 20 kis zsetonra felírjuk 1-20 -ig a számokat, és a táblázatnak megfelelően elkezdjük tologatni.

Ha mind a 20 tartalmat egyszerre tologatjuk, akkor csak belefáradunk a játékba. Ezért először csak azt figyeljük meg, hogy például az 1. memória tartalma merrefelé vándorol (a táblázat szerint): $1 \to 2 \to 5 \to 10 \to 4 \to 3 \to 1 \to 2 \to \ldots$. Vagyis pontosan 6 lépés után visszaér, majd ismét ugyanezen a *pályán* megy tovább. A táblázat alapján ellenőrizhetjük: ez azt is jelenti, hogy az 1., 2., 5., 10., 4., és 3. memóriák tartalmai is ezen a körön mozognak, és mindegyikük pontosan *hat* gombnyomásonként jut vissza eredeti helyére, vagyis 6, 12, 18, 24, ... gombnyomás után.

A többi memóriarekesz pályáit is hasonlóan deríthetjük ki, amit az alábbi ábrán szemléltethetünk:



Ez azt jelenti, hogy **6** gombnyomásonként az első körben érintett memóriák, **4** gombnyomásonként a második körben, **7** gombnyomásonként a harmadik, **2** gombnyomásonként a negyedik körben érintett memóriák kerülnek vissza eredeti helyükre, a hetes memória tartalma szerencsére mindvégig a helyén marad. Tehát azt a legkisebb számot kell megkeresnünk, amelyben a 6,4,7,2 számok mindegyike maradék nélkül megtalálható, ez pedig *legkisebb közös többszörösük*. Mivel pedig *lkkt*(6,4,7,2) = 3*4*7 = 84 és a gombot egyszer már megnyomta az automatika, ezért a Kapitánynak már csak **83** -szor kell azt megnyomnia.

Megjegyzések: i) Egy gombnyomás nyilvánvalóan a memóriatartalmak egy cserélgetése, azaz *permutációja*. A feladat megoldása a fenti ábrán alapul, amiben ezt a permutációt felbontottuk a memóriatartalmak vándorlásainak *pályáira*, amik láthatóan *diszjunkt ciklikus* permutációk (röviden ciklusok).

ii) A számítógép működése során az adatok a memóriában valóban hasonlóan, csak sajnos nem ilyen szabályosan mozognak.

iii) Ha gyerekek labdáznak, és mindegyik *csak* a „barátjának”, azaz előírás szerinti társának dobja tovább a kezébe került (akármelyik) labdát, akkor szintén felbonthatjuk a társaságot az ábra szerint „klikkekre”.

Lásd még a Döntő 1. feladatát is.

PANNON EGYETEM
MŰSZAKI INFORMATIKAI KAR

5. FELADAT: KÉRDEZZ-FELELEK

Elromlott a távolságmérő szoftvere: bármely beírt számra csak "a távolság ennél nagyobb-egyenlő?" kérdésre ad igen/nem választ. Előzetes becsléseink alapján tudjuk, hogy a következő célpont távolsága 1 és 9 billió km között van. A másodkapitány tudja, hogy a legpontosabb becslést a "felezéses" módszerrel fogja elérni: a pillanatnyilag (eddig) kiszámolt lehetséges távolság alsó és felső becslésének átlagát (számtani közepét) fogja megkérdezni a távolságmérő egységtől.

Tehát legelső kérdése az "5-nél nagyobb?" lesz. Ha erre a kérdésre az "igen" választ kapja, akkor a keresett távolság 5 és 9 közé esik, és a másodkapitány következő kérdése "7-nél nagyobb?" lesz. Ha erre a kérdésre a "nem" választ kapja, akkor a távolságról már azt tudjuk, hogy 5 és 7 közé esik, és a harmadik kérdésünk "6-nál nagyobb?" lesz.

A másodkapitány kérdéseire (a fenti hármat is beleértve) a következő válaszokat kapta:

igen, nem, nem, igen, nem, igen, igen, nem, igen, nem, nem, nem.

Ez esetben a távolságra milyen becslése (minimum és maximum) lesz a másodkapitánynak?

Beküldendő kód: A távolság alsó- és felső becslése billió km-ben:

három-három tizedesjegyre kerekítve, tizedespontokkal, pontosvesszővel elválasztva a következő alakban (összesen 11 karakter):

e.ttt;e.ttt

Vigyázz: számoláskor legalább 6 (hat) tizedesjegyre pontosan számolj, és csak a legvégén kerekíts három tizedesjegyre!

https://pexpedicio.mik.uni-pannon.hu

Az 5. feladat megoldása

A távolságot **t** -vel jelöljük. Készítsünk táblázatot:

0) Kezdet: $1 \le \mathbf{t} \le 9$.

	Kérdések		válaszok		következtetések:
1)	" 5	≤ **t** " ?	**igen**	⇒	5 ≤ **t** ≤ 9 ,
2)	" 7	≤ **t** " ?	**nem**	⇒	5 ≤ **t** ≤ 7 ,
3)	" 6	≤ **t** " ?	**nem**	⇒	5 ≤ **t** ≤ 6 ,
4)	" 5.5	≤ **t** " ?	**igen**	⇒	5.5 ≤ **t** ≤ 6 ,
5)	" 5.75	≤ **t** " ?	**nem**	⇒	5.5 ≤ **t** ≤ 5.750 ,
6)	" 5.625	≤ **t** " ?	**igen**	⇒	5.625 ≤ **t** ≤ 5.750 ,
7)	" 5.687.5	≤ **t** " ?	**igen**	⇒	5.6875 ≤ **t** ≤ 5.750 ,
8)	" 5.71875	≤ **t** " ?	**nem**	⇒	5.6875 ≤ **t** ≤ 5.71875 ,
9)	" 5.703125	≤ **t** " ?	**igen**	⇒	5.703125 ≤ **t** ≤ 5.71875 ,
10)	" 5.7109375	≤ **t** " ?	**nem**	⇒	5.703125 ≤ **t** ≤ 5.7109375 ,
11)	" 5.70703125	≤ **t** " ?	**nem**	⇒	5.703125 ≤ **t** ≤ 5.70703125 ,
12)	" 5.705078125	≤ **t** " ?	**nem**	⇒	5.703125 ≤ **t** ≤ 5.705078125 .

Tehát a feltöltendő kód: "`5.703;5.705`"

Megjegyzés: Az "*intervallum-felezéses*" módszer valóban *nagyon gyors, egyszerű és általános* megoldási módszer, *bármilyen* $f(x)=0$ egyenlet nagy pontosságú, közelítő megoldására, ahol $f(x)$ bármilyen folytonos függvény. n lépés után az ismeretlen x^* gyök olyan két határ, a_n és b_n közé szorítható, vagyis $a_n \le x^* \le b_n$, amelyekre távolsága $b_n - a_n \le (b-a)/2^n$, vagyis a közelítés hibája $\varepsilon \le (b-a)/2^n$, ahol [a,b] a kezdő intervallum. Tehát kb. 3 lépésenként (pontosabban 3.01 lépésenként) van egy újabb pontos tizedesjegyünk (mert $2^{3.01} \approx 10$).

A kitűzött feladatban $n = 12$ lépés után a kapott (utolsó) intervallum középpontja (felezőpontja):

$x_{n+1} = 1+ 4*\left(1+\Sigma_i (-1)^{\text{válasz}} \cdot 1/2^i\right) = 1 +$

$+4*\big(1+(1/2)-(1/4)-(1/8)+(1/16)-(1/32)+(1/64)+(1/128)-(1/256)+(1/512)-(1/1024)-(1/2048)-(1/4096)\big) =$

$= \underline{5.7041015625}$,

az utolsó intervallum átmérője (hossza) $r_{n+1} = 4000/2^n = 4000/4096 =$ `0.9765625` ,

tehát az utolsó intervallum végpontjai (minimuim, maximum):

`min` $= x_{n+1} - r_{n+1} = 5704.1015625 - 0.9765625 =$ `5703.125`

`max` $= x_{n+1} + r_{n+1} = 5704.1015625 + 0.9765625 =$ `5705.078125` .

PANNON EGYETEM

MŰSZAKI INFORMATIKAI KAR

6. FELADAT: REJTVÉNY

A csillagközi térben a hosszú, eseménytelen száguldás alatt az utasok „számítógépes” rejtvényekkel szórakoztatják egymást. Egy hosszú „kockás” papírra („szalagra”) ceruzával felírták a rejtjelezett szöveget, minden betűt egy-egy négyzetbe, például: WIHIKRIPAI , a szó előtt és után sok üres négyzet van. A betűket egyesével kell törölni és újraírni, a következő szabályok szerint. Az író-törlő ceruzának öt „állapota” van: 0,1,2,3,4, induláskor a 0 állapotban van és a W betűre mutat, a 4 állapotban vége a játéknak, megáll a ceruza.

A ceruza mozgását és működését az alábbi 64 utasítás irányítja:

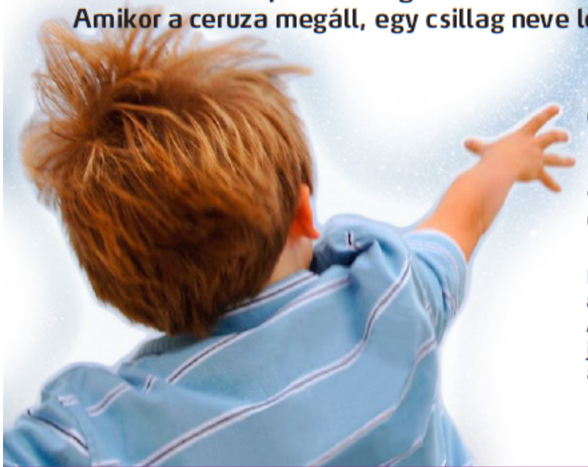
[01]:(0,A)->(1,Z,+1)	[17]:(1,A)->(1,A,+1)	[33]:(2,A)->(2,A,-1)	[49]:(3,A)->(1,S,+1)
[02]:(0,B)->(1,Z,+1)	[18]:(1,B)->(1,B,+1)	[34]:(2,B)->(3,B,+1)	[50]:(3,B)->(1,B,+1)
[03]:(0,E)->(1,Z,+1)	[19]:(1,E)->(1,E,+1)	[35]:(2,E)->(3,E,+1)	[51]:(3,E)->(1,E,+1)
[04]:(0,G)->(1,Z,+1)	[20]:(1,G)->(1,G,+1)	[36]:(2,G)->(3,G,+1)	[52]:(3,G)->(1,G,+1)
[05]:(0,H)->(1,Z,+1)	[21]:(1,H)->(1,H,+1)	[37]:(2,H)->(2,H,-1)	[53]:(3,H)->(1,T,+1)
[06]:(0,I)->(1,Z,+1)	[22]:(1,I)->(1,I,+1)	[38]:(2,I)->(2,I,-1)	[54]:(3,I)->(1,E,+1)
[07]:(0,K)->(1,Z,+1)	[23]:(1,K)->(1,K,+1)	[39]:(2,K)->(2,K,-1)	[55]:(3,K)->(1,L,+1)
[08]:(0,L)->(1,Z,+1)	[24]:(1,L)->(1,L,+1)	[40]:(2,L)->(3,L,+1)	[56]:(3,L)->(1,L,+1)
[09]:(0,P)->(1,Z,+1)	[25]:(1,P)->(1,P,+1)	[41]:(2,P)->(2,P,-1)	[57]:(3,P)->(1,U,+1)
[10]:(0,R)->(1,Z,+1)	[26]:(1,R)->(1,R,+1)	[42]:(2,R)->(2,R,-1)	[58]:(3,R)->(1,G,+1)
[11]:(0,S)->(1,Z,+1)	[27]:(1,S)->(1,S,+1)	[43]:(2,S)->(3,S,+1)	[59]:(3,S)->(1,S,+1)
[12]:(0,T)->(1,Z,+1)	[28]:(1,T)->(1,T,+1)	[44]:(2,T)->(3,T,+1)	[60]:(3,T)->(1,T,+1)
[13]:(0,U)->(1,Z,+1)	[29]:(1,U)->(1,U,+1)	[45]:(2,U)->(3,U,+1)	[61]:(3,U)->(1,U,+1)
[14]:(0,W)->(1,B,+1)	[30]:(1,W)->(1,W,+1)	[46]:(2,W)->(2,W,-1)	[62]:(3,W)->(1,B,+1)
[15]:(0,Z)->(1,Z,+1)	[31]:(1,Z)->(1,Z,+1)	[47]:(2,Z)->(2,Z,-1)	[63]:(3,Z)->(1,Z,+1)
[16]:(0,_)->(1,_, 0)	[32]:(1,_)->(2,_,-1)	[48]:(2,_)->(2,_,-1)	[64]:(3,_)->(4,_, 0)

ahol az [s]:(a,b)->(c,d,e) alakú utasítások jelentése a következő: „s” a sorszám, „a” a ceruza állapota, „b” betűre mutat, „c” állapotba fog „átbillenni”, „d” betűre fogja kicserélni „b”-t, és „e” -nek megfelelő irányba fog továbblépni: e= +1 esetén jobbra, e= -1 esetén balra egy betűt, e=0 esetén pedig semerre.

„_” az üres négyzet jele, „b”=„d” esetén a betű nem változik a négyzetben.

Tehát minden lépésben meg kell keresned, melyik utasítás mozgatja tovább a ceruzát!

Amikor a ceruza megáll, egy csillag neve lesz a papíron.



Beküldendő kód: a csillag neve, és melyik csillagképben található,

nagybetűkkel írva, pontosvesszővel elválasztva, a következő alakban: c...c;k...k ahol c...c a csillag neve és k...k a csillagkép neve.

Légy türelmes: A ceruza sokáig semmi érdekeset nem csinál, a megfejtés több, mint száz lépésig tart! A ceruza állapotát egy dobókockával jelölheted (a 0 állapotban dugd el a kockát).

https://pexpedicio.mik.uni-pannon.hu

A 6. feladat megoldása

Megadjuk a (papír)szalag, a "fej" (ceruza) ^ és a gép állapotát ( )-ben, baloldalon pedig [ ] -ben a végrehajtott utasítás sorszámát.

```
utasítás              gép    lépés  Megj.
sorsz.             állapota  sorsz.      .
     ..._WIHIKRIPAI_...
         ^              (0)
[14] ..._BIHIKRIPAI_...        1.lépés B<>W
          ^             (1)
[22] ..._BIHIKRIPAI_...        2.lépés
           ^            (1)
[21] ..._BIHIKRIPAI_...        3.lépés
            ^           (1)
[22] ..._BIHIKRIPAI_...        4.lépés
             ^          (1)
[23] ..._BIHIKRIPAI_...        5.lépés
              ^         (1)
[26] ..._BIHIKRIPAI_...        6.lépés
               ^        (1)
[22] ..._BIHIKRIPAI_...        7.lépés
                ^       (1)
[25] ..._BIHIKRIPAI_...        8.lépés
                 ^      (1)
[17] ..._BIHIKRIPAI_...        9.lépés
                  ^     (1)
[22] ..._BIHIKRIPAI_...       10.lépés
                   ^    (1)
[32] ..._BIHIKRIPAI_...       11.lépés
                  ^     (2)
[38] ..._BIHIKRIPAI_...       12.lépés
                 ^      (2)
[33] ..._BIHIKRIPAI_...       13.lépés
                ^       (2)
[41] ..._BIHIKRIPAI_...       14.lépés
               ^        (2)
[38] ..._BIHIKRIPAI_...       15.lépés
              ^         (2)
[42] ..._BIHIKRIPAI_...       16.lépés
             ^          (2)
[39] ..._BIHIKRIPAI_...       17.lépés
            ^           (2)
[38] ..._BIHIKRIPAI_...       18.lépés
           ^            (2)
[37] ..._BIHIKRIPAI_...       19.lépés
          ^             (2)
[38] ..._BIHIKRIPAI_...       20.lépés
         ^              (2)
[34] ..._BIHIKRIPAI_...       21.lépés
          ^             (3)
[54] ..._BEHIKRIPAI_...       22.lépés E<>I
           ^            (1)
[21] ..._BEHIKRIPAI_...       23.lépés
            ^           (1)
[22] ..._BEHIKRIPAI_...       24.lépés
             ^          (1)
[23] ..._BEHIKRIPAI_...       25.lépés
              ^         (1)
[26] ..._BEHIKRIPAI_...       26.lépés
               ^        (1)
[22] ..._BEHIKRIPAI_...       27.lépés
                ^       (1)
[25] ..._BEHIKRIPAI_...       28.lépés
                 ^      (1)
[17] ..._BEHIKRIPAI_...       29.lépés
                  ^     (1)
[22] ..._BEHIKRIPAI_...       30.lépés
                   ^    (1)
[32] ..._BEHIKRIPAI_...       31.lépés
                  ^     (2)
[38] ..._BEHIKRIPAI_...       32.lépés
                 ^      (2)
[33] ..._BEHIKRIPAI_...       33.lépés
                ^       (2)
[41] ..._BEHIKRIPAI_...       34.lépés
               ^        (2)
[38] ..._BEHIKRIPAI_...       35.lépés
              ^         (2)
[42] ..._BEHIKRIPAI_...       36.lépés
             ^          (2)
[39] ..._BEHIKRIPAI_...       37.lépés
            ^           (2)
[38] ..._BEHIKRIPAI_...       38.lépés
           ^            (2)
[32] ..._BEHIKRIPAI_...       39.lépés
          ^             (2)
[35] ..._BEHIKRIPAI_...       40.lépés
           ^            (3)
[53] ..._BETIKRIPAI_...       41.lépés T<>H
            ^           (1)
[22] ..._BETIKRIPAI_...       42.lépés
             ^          (1)
[23] ..._BETIKRIPAI_...       43.lépés
              ^         (1)
[26] ..._BETIKRIPAI_...       44.lépés
               ^        (1)
[22] ..._BETIKRIPAI_...       45.lépés
                ^       (1)
[25] ..._BETIKRIPAI_...       46.lépés
                 ^      (1)
[17] ..._BETIKRIPAI_...       47.lépés
                  ^     (1)
[22] ..._BETIKRIPAI_...       48.lépés
                   ^    (1)
[32] ..._BETIKRIPAI_...       49.lépés
                  ^     (2)
[38] ..._BETIKRIPAI_...       50.lépés
                 ^      (2)
[33] ..._BETIKRIPAI_...       51.lépés
                ^       (2)
[41] ..._BETIKRIPAI_...       52.lépés
               ^        (2)
[38] ..._BETIKRIPAI_...       53.lépés
              ^         (2)
[42] ..._BETIKRIPAI_...       54.lépés
             ^          (2)
[39] ..._BETIKRIPAI_...       55.lépés
            ^           (2)
[38] ..._BETIKRIPAI_...       56.lépés
           ^            (2)
[44] ..._BETIKRIPAI_...       57.lépés
            ^           (3)
[54] ..._BETEKRIPAI_...       58.lépés E<>I
             ^          (1)
[23] ..._BETEKRIPAI_...       59.lépés
              ^         (1)
[26] ..._BETEKRIPAI_...       60.lépés
               ^        (1)
[22] ..._BETEKRIPAI_...       61.lépés
                ^       (1)
[25] ..._BETEKRIPAI_...       62.lépés
                 ^      (1)
[17] ..._BETEKRIPAI_...       63.lépés
                  ^     (1)
[22] ..._BETEKRIPAI_...       64.lépés
                   ^    (1)
[32] ..._BETEKRIPAI_...       65.lépés
                  ^     (2)
```

```
[38] ..._BETEKRIPAI_...       66.lépés
                 ^      (2)
[33] ..._BETEKRIPAI_...       67.lépés
                ^       (2)
[41] ..._BETEKRIPAI_...       68.lépés
               ^        (2)
[38] ..._BETEKRIPAI_...       69.lépés
              ^         (2)
[42] ..._BETEKRIPAI_...       70.lépés
             ^          (2)
[39] ..._BETEKRIPAI_...       71.lépés
            ^           (2)
[35] ..._BETEKRIPAI_...       72.lépés
             ^          (3)
[55] ..._BETELRIPAI_...       73.lépés L<>K
              ^         (1)
[26] ..._BETELRIPAI_...       74.lépés
               ^        (1)
[22] ..._BETELRIPAI_...       75.lépés
                ^       (1)
[25] ..._BETELRIPAI_...       76.lépés
                 ^      (1)
[17] ..._BETELRIPAI_...       77.lépés
                  ^     (1)
[22] ..._BETELRIPAI_...       78.lépés
                   ^    (1)
[32] ..._BETELRIPAI_...       79.lépés
                  ^     (2)
[38] ..._BETELRIPAI_...       80.lépés
                 ^      (2)
[33] ..._BETELRIPAI_...       81.lépés
                ^       (2)
[41] ..._BETELRIPAI_...       82.lépés
               ^        (2)
[38] ..._BETELRIPAI_...       83.lépés
              ^         (2)
[42] ..._BETELRIPAI_...       84.lépés
             ^          (2)
[40] ..._BETELRIPAI_...       85.lépés
              ^         (3)
[58] ..._BETELGIPAI_...       86.lépés G<>R
               ^        (1)
[22] ..._BETELGIPAI_...       87.lépés
                ^       (1)
[25] ..._BETELGIPAI_...       88.lépés
                 ^      (1)
[17] ..._BETELGIPAI_...       89.lépés
                  ^     (1)
[22] ..._BETELGIPAI_...       90.lépés
                   ^    (1)
[32] ..._BETELGIPAI_...       91.lépés
                  ^     (2)
[38] ..._BETELGIPAI_...       92.lépés
                 ^      (2)
[33] ..._BETELGIPAI_...       93.lépés
                ^       (2)
[41] ..._BETELGIPAI_...       94.lépés
               ^        (2)
```

```
[38] ..._BETELGIPAI_...       95.lépés
              ^         (2)
[36] ..._BETELGIPAI_...       96.lépés
               ^        (3)
[54] ..._BETELGEPAI_...       97.lépés E<>I
                ^       (1)
[25] ..._BETELGEPAI_...       98.lépés
                 ^      (1)
[17] ..._BETELGEPAI_...       99.lépés
                  ^     (1)
[22] ..._BETELGEPAI_...      100.lépés
                   ^    (1)
[32] ..._BETELGEPAI_...      101.lépés
                  ^     (2)
[38] ..._BETELGEPAI_...      102.lépés
                 ^      (2)
[33] ..._BETELGEPAI_...      103.lépés
                ^       (2)
[41] ..._BETELGEPAI_...      104.lépés
               ^        (2)
[35] ..._BETELGEPAI_...      105.lépés
                ^       (3)
[57] ..._BETELGEUAI_...      106.lépés U<>P
                 ^      (1)
[17] ..._BETELGEUAI_...      107.lépés
                  ^     (1)
[22] ..._BETELGEUAI_...      108.lépés
                   ^    (1)
[32] ..._BETELGEUAI_...      109.lépés
                  ^     (2)
[38] ..._BETELGEUAI_...      110.lépés
                 ^      (2)
[33] ..._BETELGEUAI_...      111.lépés
                ^       (2)
[45] ..._BETELGEUAI_...      112.lépés
                 ^      (3)
[49] ..._BETELGEUSI_...      113.lépés S<>A
                  ^     (1)
[22] ..._BETELGEUSI_...      114.lépés
                   ^    (1)
[32] ..._BETELGEUSI_...      115.lépés
                  ^     (2)
[38] ..._BETELGEUSI_...      116.lépés
                 ^      (2)
[43] ..._BETELGEUSI_...      117.lépés
                  ^     (3)
[54] ..._BETELGEUSE_...      118.lépés E<>I
                   ^    (1)
[32] ..._BETELGEUSE_...      119.lépés
                  ^     (2)
[35] ..._BETELGEUSE_...      120.lépés
                   ^    (3)
[64] ..._BETELGEUSE_...      121.lépés
                   ^    (4) stop
```

tehát a válasz: BETELGEUSE az ORION csillagképben.

Megjegyzés: Sajnos a feladat eléggé hosszadalmasra és unalmasra sikerült, de egy csillagközi utazáson bőven van idő ilyesmire!

Bármennyire hihetetlen első hallásra: a fenti ceruza és radír (és a kockás papír) egy számítógépet, a *Turing-gépet* szimulálja, vagyis *minden* számítógépet (!), annak elve szerint működik. Hiszen mindegyik számítógép ír-olvas-töröl valahonnan (RAM, merevlemez, pendrive, stb.), lépteti az író-olvasó "fejét" ide-oda, és mindezt a számítógépben éppen akkor tárolt rengeteg adat, utasítás, program, feltétel (amiket röviden "állapotnak" nevezünk) hatására.

Nos, bár ezen feltételek rengetegen vannak, talán még senki sem sorolta fel őket, *de* fel lehetne sorolni 1-től M -ig (persze M nagyon nagy szám). Ez utóbbi volt Turing angol matematikus (1912-1954) ötlete: elég csak annyit tudnunk, hogy a számítógépnek valamilyen "belső" állapotai vannak, mondjuk $q_1 , \dots , q_M$. A dobókocka csak azért kellett a feladatban, hogy mi (rejtvényfejtő emberek) el ne felejtsük, hogy éppen melyik állapotban van a számítógép. A feladatban [1] és [64] között felsorolt `[s]:(a,b)->(c,d,e)` alakú utasítások maga a program, ami a számítógépet vezérli: *ha* éppen "ez vagy az" van, pontosabban `(a,b)` van, *akkor* legyen a következő lépésben "amaz", vagyis `(c,d,e)` .
(Részletesebben lásd pl. Szalkai [2001] könyvében.)

PANNON EGYETEM
MŰSZAKI INFORMATIKAI KAR

7. FELADAT: A JELSZÓ

Űrhajósaink már felvették a kapcsolatot a megtalált ősök leszármazottaival, már csak pár millió km van hátra. Azonban, biztonsági okokból az utódok meg akarnak győződni, hogy valóban őseik leszármazottai látogatják meg őket, ezért jelszót kérnek tőlük, az alábbi rejtvényes formában.

i) **a Föld térképén a (47°05'21.8"N ; 17°54'26.9"E) helyen levő, legnagyobb felirattal, (kék háttérrel) megjelenő intézmény nevének kezdőbetűi,**

ii) **a 3^{2019} hatvány utolsó számjegye,**

iii) **Victor Hugo egyik regényében Jean Valjean rabruháján található ötjegyű szám *osztóinak száma.***

Beküldendő kód: A megfejtés xxx_y_zzz alakban,

ahol xxx az i) feladat megoldása (három betű), y a ii) feladat megoldása (egy számjegy), zzz pedig a iii) feladat megoldása (egy szám, nem feltétlenül három jegyű, a száznál kisebb számok 0-val kezdődnek), a megfejtések között kettő aláhúzás karakter van).

https://pexpedicio.mik.uni-pannon.hu

A 7. feladat megoldása

i) Az internet segítségével könnyen megtalálhatjuk a Pannon Egyetem **M**űszaki **I**nformatikai **K**arát = MIK a térképen, tehát a megfejtés: **MIK** .

Egy kis gyakorlás a hatvanas számrendszerben:
47°05'21.8" = 47+ 5/60+21,8/3600 ≈ **47.089389** ,
17°54'26.9" = 17+54/60+26,9/3600 ≈ **17.907472** .

ii) A 3^n hatványok (n=0,1,...) utolsó jegyei: 1,3,9,7, 1,3,9,7 ... , tehát négyesével *ismétlődnek*, például 3^4 utolsó jegye 1. Az „*utolsó jegye*” szavakat rövidítsük a ≡ jellel. Mivel 2019 = 4*504+3, ezért

$$3^{2019} \equiv 3^{4*504+3} \equiv (3^4)^{504} * 3^3 \equiv (1)^{504} * 7 \equiv \mathbf{7}\,.$$

Tehát a megfejtés: **7** .

iii) Könnyen kinyomozhatjuk, hogy Victor Hugo: *A nyomorultak* c. regényéről van szó, még a polcról sem kell leemelnünk: http://mek.oszk.hu/10500/10546/index.phtml . Aprólékos kereséssel megtaláljuk a bűvös számot és annak prímtényezős felbontását: 24601 = 73 * 337 , aminek tehát a következő osztói vannak: 1, 73, 337 és 24601. Tehát a megfejtés: **4** .

A beküldendő kód tehát: **MIK_7_004** .

Megjegyzések: A harmadik kérdésnél nem hangsúlyoztuk, hogy csak valódi vagy az összes osztó számát kérdeztük. Ha egyik megkülönböztetés sincs feltüntetve, akkor az *összes* osztóra gondol mindenki.

Ha angolul a "*jean valjean number*" kulcsra keresünk, akkor a következő érdekes lapokat találhatjuk: https://en.wikipedia.org/wiki/24,601 (matematikai lap),
https://en.wikipedia.org/wiki/Jean_Valjean és
https://en.wikipedia.org/wiki/24601_Valjean =>
https://en.wikipedia.org/wiki/List_of_minor_planets:_24001%E2%80%9325000#601
(kisbolygók listája). Tehát a 24601 sorszámú kisbolygó is Jean Valjean nevét viseli.

8. FELADAT: AZ AJÁNDÉKOK

A Kapitány készülődik a leszálláshoz, az új naprendszer bolygói között kell navigálnia. Eközben a legénység már csomagolja is az ajándékokat a fogadó bolygó leggyorsabban nyakukba ugró gyermekeinek.

Mindegyik gyermek pontosan 111 dkg ajándékot fog kapni, de az űrhajón csak 2, 3 és 11 dkg méretű díszdobozkák vannak. Hányféle ajándékcsomagot tudnak készíteni, vagyis hányféleképpen lehet 111 dkg-ot szétosztani 2, 3 és 11 dkg méretű dobozkákba? (A dobozok sorrendje lényegtelen, vagyis csak az a lényeg, hogy hány 2 dkg, hány 3 dkg és hány 11 dkg méretű dobozkát kap az ajándékozott).

**Beküldendő kód:
Egyetlen szám: a dobozkákba csomagolások lehetséges száma.**

(Megjegyzés: a fogadó bolygó többi (lassabb) gyermekei is pontosan 111 dkg ajándékot fognak kapni.)

SZÉCHENYI 2020
Magyarország Kormánya
Európai Unió
Európai Szociális Alap
BEFEKTETÉS A JÖVŐBE

A 8. feladat megoldása

i) megoldás: felírjuk az összes lehetőséget. Mivel elég sok van, aszerint rendeztük őket, hogy hány *nagy* dobozt használtunk fel.

```
  1.  11*  0 +   2*  0 +   3* 37 = 111
  2.  11*  0 +   2*  3 +   3* 35 = 111
  3.  11*  0 +   2*  6 +   3* 33 = 111
  4.  11*  0 +   2*  9 +   3* 31 = 111
  5.  11*  0 +   2* 12 +   3* 29 = 111
  6.  11*  0 +   2* 15 +   3* 27 = 111
  7.  11*  0 +   2* 18 +   3* 25 = 111
  8.  11*  0 +   2* 21 +   3* 23 = 111
  9.  11*  0 +   2* 24 +   3* 21 = 111
 10.  11*  0 +   2* 27 +   3* 19 = 111
 11.  11*  0 +   2* 30 +   3* 17 = 111
 12.  11*  0 +   2* 33 +   3* 15 = 111
 13.  11*  0 +   2* 36 +   3* 13 = 111
 14.  11*  0 +   2* 39 +   3* 11 = 111
 15.  11*  0 +   2* 42 +   3*  9 = 111
 16.  11*  0 +   2* 45 +   3*  7 = 111
 17.  11*  0 +   2* 48 +   3*  5 = 111
 18.  11*  0 +   2* 51 +   3*  3 = 111
 19.  11*  0 +   2* 54 +   3*  1 = 111
-----------------------------------------
 20.  11*  1 +   2*  2 +   3* 32 = 111
 21.  11*  1 +   2*  5 +   3* 30 = 111
 22.  11*  1 +   2*  8 +   3* 28 = 111
 23.  11*  1 +   2* 11 +   3* 26 = 111
 24.  11*  1 +   2* 14 +   3* 24 = 111
 25.  11*  1 +   2* 17 +   3* 22 = 111
 26   11*  1 +   2* 20 +   3* 20 = 111
 27   11*  1 +   2* 23 +   3* 18 = 111
 28   11*  1 +   2* 26 +   3* 16 = 111
 29   11*  1 +   2* 29 +   3* 14 = 111
 30   11*  1 +   2* 32 +   3* 12 = 111
 31   11*  1 +   2* 35 +   3* 10 = 111
 32   11*  1 +   2* 38 +   3*  8 = 111
 33   11*  1 +   2* 41 +   3*  6 = 111
 34   11*  1 +   2* 44 +   3*  4 = 111
 35   11*  1 +   2* 47 +   3*  2 = 111
 36   11*  1 +   2* 50 +   3*  0 = 111
-----------------------------------------
 37   11*  2 +   2*  1 +   3* 29 = 111
 38   11*  2 +   2*  4 +   3* 27 = 111
 39   11*  2 +   2*  7 +   3* 25 = 111
 40   11*  2 +   2* 10 +   3* 23 = 111
 41   11*  2 +   2* 13 +   3* 21 = 111
 42   11*  2 +   2* 16 +   3* 19 = 111
 43   11*  2 +   2* 19 +   3* 17 = 111
 44   11*  2 +   2* 22 +   3* 15 = 111
 45   11*  2 +   2* 25 +   3* 13 = 111
 46   11*  2 +   2* 28 +   3* 11 = 111
 47   11*  2 +   2* 31 +   3*  9 = 111
 48   11*  2 +   2* 34 +   3*  7 = 111
 49   11*  2 +   2* 37 +   3*  5 = 111
 50   11*  2 +   2* 40 +   3*  3 = 111
 51   11*  2 +   2* 43 +   3*  1 = 111
-----------------------------------------
 52   11*  3 +   2*  0 +   3* 26 = 111
 53   11*  3 +   2*  3 +   3* 24 = 111
 54   11*  3 +   2*  6 +   3* 22 = 111
 55   11*  3 +   2*  9 +   3* 20 = 111
 56   11*  3 +   2* 12 +   3* 18 = 111
 57   11*  3 +   2* 15 +   3* 16 = 111
 58   11*  3 +   2* 18 +   3* 14 = 111
 59   11*  3 +   2* 21 +   3* 12 = 111
 60   11*  3 +   2* 24 +   3* 10 = 111
 61   11*  3 +   2* 27 +   3*  8 = 111
 62   11*  3 +   2* 30 +   3*  6 = 111
 63   11*  3 +   2* 33 +   3*  4 = 111
 64   11*  3 +   2* 36 +   3*  2 = 111
 65   11*  3 +   2* 39 +   3*  0 = 111
-----------------------------------------
 66   11*  4 +   2*  2 +   3* 21 = 111
 67   11*  4 +   2*  5 +   3* 19 = 111
 68   11*  4 +   2*  8 +   3* 17 = 111
 69   11*  4 +   2* 11 +   3* 15 = 111
 70   11*  4 +   2* 14 +   3* 13 = 111
 71   11*  4 +   2* 17 +   3* 11 = 111
 72   11*  4 +   2* 20 +   3*  9 = 111
 73   11*  4 +   2* 23 +   3*  7 = 111
 74   11*  4 +   2* 26 +   3*  5 = 111
 75   11*  4 +   2* 29 +   3*  3 = 111
 76   11*  4 +   2* 32 +   3*  1 = 111
-----------------------------------------
 77   11*  5 +   2*  1 +   3* 18 = 111
 78   11*  5 +   2*  4 +   3* 16 = 111
 79   11*  5 +   2*  7 +   3* 14 = 111
 80   11*  5 +   2* 10 +   3* 12 = 111
 81   11*  5 +   2* 13 +   3* 10 = 111
 82   11*  5 +   2* 16 +   3*  8 = 111
 83   11*  5 +   2* 19 +   3*  6 = 111
 84   11*  5 +   2* 22 +   3*  4 = 111
 85   11*  5 +   2* 25 +   3*  2 = 111
 86   11*  5 +   2* 28 +   3*  0 = 111
-----------------------------------------
 87   11*  6 +   2*  0 +   3* 15 = 111
 88   11*  6 +   2*  3 +   3* 13 = 111
 89   11*  6 +   2*  6 +   3* 11 = 111
 90   11*  6 +   2*  9 +   3*  9 = 111
 91   11*  6 +   2* 12 +   3*  7 = 111
 92   11*  6 +   2* 15 +   3*  5 = 111
 93   11*  6 +   2* 18 +   3*  3 = 111
 94   11*  6 +   2* 21 +   3*  1 = 111
-----------------------------------------
 95   11*  7 +   2*  2 +   3* 10 = 111
 96   11*  7 +   2*  5 +   3*  8 = 111
 97   11*  7 +   2*  8 +   3*  6 = 111
 98   11*  7 +   2* 11 +   3*  4 = 111
 99   11*  7 +   2* 14 +   3*  2 = 111
100   11*  7 +   2* 17 +   3*  0 = 111
-----------------------------------------
101   11*  8 +   2*  1 +   3*  7 = 111
102   11*  8 +   2*  4 +   3*  5 = 111
103   11*  8 +   2*  7 +   3*  3 = 111
104   11*  8 +   2* 10 +   3*  1 = 111
-----------------------------------------
105   11*  9 +   2*  0 +   3*  4 = 111
106   11*  9 +   2*  3 +   3*  2 = 111
107   11*  9 +   2*  6 +   3*  0 = 111
-----------------------------------------
```

Tehát a válasz: <u>107 - *féleképpen*</u> .

ii) megoldás: Próbáljuk meg a feladatot visszavezetni kisebb feladatokra. Jelöljük ezért a_n -el azt a számot, ahány féleképpen el lehet helyezni n dkg ajándékot az adott méretű dobozokba. Tehát általában kérdezzük: $a_n = ?$

Több eset is lehetséges. Megtöltünk például egy 2 dkg -s dobozt, marad n-2 dkg csomagolni valónk, ami a_{n-2} lehetőséget jelent. De megtölthetünk (először) egy 3 vagy egy 11 dkg -s dobozt, és a maradék n-3 illetve n-11 dkg ajándékot kell dobozolnunk, ami a_{n-3} és a_{n-11} lehetőséget jelent. Eddig tehát $a_n = a_{n-2} + a_{n-3} + a_{n-11}$.

Azonban néhány esetet többször számoltunk! Ha például egy 2 és egy 3 dkg -s dobozt is megtöltöttünk, akkor a maradék n-5 dkg csomagolásának a_{n-5} esetét kétszer számoltuk, tehát egyszer le kell vonnunk az előző képletből. Hasonlóan egy-egy 2 és 11 dkg doboz megtöltése után maradt a_{n-13} , és a megtöltött 3 és 11 dkg dobozok után maradt a_{n-14} eseteket is kétszer számoltuk, ezeket is le kell vonnunk, tehát eddig $a_n = a_{n-2} + a_{n-3} + a_{n-11} - a_{n-5} - a_{n-13} - a_{n-14}$.

Mi van azzal az esettel, amikor mind a három típusú dobozból egyet-egyet megtöltöttünk, és maradt n-(2+3+11) = n-16 dkg ? Bizony, ezeket az eseteket az első bekezdés mindegyik tagjában beleszámoltuk, majd a második bekezdés mindegyik tagjában kivontuk, vagyis végeredményben +3-3=0 -szor számoltuk, vagyis kihagytuk! Pótolnunk kell! (Most az úgynevezett *logikai szitát* használtuk.)

Ezért a végleges összefüggés:

$$a_n = a_{n-2} + a_{n-3} + a_{n-11} - a_{n-5} - a_{n-13} - a_{n-14} + a_{n-16} \,, \qquad (*)$$

ami minden 16 -nál nagyobb n -re igaz.

Már csak ki kell számolnunk az első 16 esetet, és a fenti összefüggés máris megadja az összes a_n szám értékét, szépen egyesével:

n =	0	1	2	3	4	5	6	7	8	9
a_n =	1	**0**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**2**
n =	10	11	12	13	14	15	16			
a_n =	**2**	**3**	**3**	**3**	**4**	**4**	**4**			

(**)

továbbá:

a_{17} = 4 + 4 + 2 - 3 - 1 - 1 + 0 = 5 ,
a_{18} = 4 + 4 + 1 - 3 - 1 - 1 + 1 = 5 ,
. . .
a_{111} = 103+102+88 -98-85-83 +80 = **107** .

Megjegyzések: A fenti (*), úgynevezett *rekurzív összefüggés* és a (*) *kezdeti értékek* alapján már könnyen és gyorsan kiszámolhatjuk az a_n sorozat bármelyik elemét. Egyetlen hátránya, hogy bármelyik a_n kiszámításához előbb ki kell számolnunk az *összes* megelőző tagot, vagyis például a_{1000} kiszámolása előtt 1000 megelőző tag (a_1, a_2 , ... , a_{999}) ismerete szükséges. Ez megint a *dominóelv*: a felállított dominók közül csak akkor dől le a legutolsó, ha az összes megelőző hiánytalanul, "hibátlanul" feldőlt.

Van ugyan olyan módszer (képlet) a_n -re, amely azonnal megadja n ismeretében a_n értékét, bármelyik n természetes számra, de még a végeredmény sem fér bele ebbe a kis füzetbe. (Az általános módszert lásd pl. Szalkai [2001] könyvében.)

A (**) -hoz hasonló rekurzív összefüggések és az *önmagukat meghívó* (felhasználó), ún. rekurzív program-rutinok fontos szerepet játszanak a számítógép programokban!

Az alábbi megoldás még sok egyetemi hallgatónak is nehéz dió.

iii) megoldás: (generátorfüggvénnyel)

Tétel ("Pénzváltási probléma"): Ha $h_1, \ldots, h_k \in \mathrm{N}$ tetszőleges rögzített pozitív számok, $k \in \mathrm{N}$ tetszőleges, és tetszőleges $n \in \mathrm{N}$ esetén a_n jelöli a

$$h_1 y_1 + \ldots + h_k y_k = n$$

"Diophantikus" egyenlet nemnegatív megoldásainak számát, akkor az (a_n) sorozat generátor-függvénye

$$F(x) = \sum_{n=0}^{\infty} a_n x^n = \frac{1}{(1-x^{h_1}) \ldots (1-x^{h_k})}$$

(Lásd Szalkai [2001] könyvében)

A 8. feladat a $2y_1 + 3y_2 + 11y_3 = 111$ egyenletet jelenti, tehát nekünk a

$$\frac{1}{(1-x^2)(1-x^3)(1-x^{11})}$$

függvény Taylor sorfejtésének 111 -dik tagja kell. A számolásokat nem részletezve:

$\frac{1}{(1-x^2)(1-x^3)(1-x^{11})} = 1+x^2+x^3+x^4+x^5+2x^6+x^7+2x^8+2x^9+2x^{10}+3x^{11}+3x^{12}+3x^{13}+4x^{14}+4x^{15} + 4x^{16}+5x^{17}+5x^{18}+5x^{19}+6x^{20}+6x^{21}+7x^{22}+7x^{23}+8x^{24}+8x^{25}+9x^{26}+9x^{27}+10x^{28}+10x^{29}+11x^{30}+11x^{31} + 12x^{32}+13x^{33}+13x^{34}+14x^{35}+15x^{36}+15x^{37}+16x^{38}+17x^{39}+17x^{40}+18x^{41}+19x^{42}+19x^{43}+21x^{44}+21x^{45} + 22x^{46}+23x^{47}+24x^{48}+24x^{49}+26x^{50}+26x^{51}+27x^{52}+28x^{53}+29x^{54}+30x^{55}+31x^{56}+32x^{57}+33x^{58}+34x^{59} + 35x^{60}+36x^{61}+37x^{62}+38x^{63}+39x^{64}+40x^{65}+42x^{66}+42x^{67}+44x^{68}+45x^{69}+46x^{70}+47x^{71}+49x^{72}+49x^{73} + 51x^{74}+52x^{75}+53x^{76}+55x^{77}+56x^{78}+57x^{79}+59x^{80}+60x^{81}+61x^{82}+63x^{83}+64x^{84}+65x^{85}+67x^{86}+68x^{87} + 70x^{88}+71x^{89}+73x^{90}+74x^{91}+76x^{92}+77x^{93}+79x^{94}+80x^{95}+82x^{96}+83x^{97}+85x^{98}+87x^{99}+88x^{100}+90x^{101} +92x^{102}+93x^{103}+95x^{104}+97x^{105}+98x^{106}+100x^{107}+102x^{108}+103x^{109}+106x^{110}+ \underline{\mathbf{107x^{111}}} +109x^{112}+ 111x^{113}+113x^{114}+114x^{115}+117x^{116}+118x^{117}+120x^{118}+122x^{119} + \ldots$

így a válasz: 107 .

Megjegyzés: A fenti sorfejtésben minden $n \le 119$ értékre is látjuk a választ, a_n értékét (hol?) .

A Döntő feladatai

A Döntő feladatai

D1. Feladat: /10 pont/

A focimeccs előtt bemelegít a válogatott. Már rajtuk van a mez 1-től 11-ig, mindegyik játékosnál a saját labdája van (piros, sárga, ...), egy nagy körben állnak. Az edző utasítása a következő: minden sípszóra minden játékos az éppen nála levő labdát (akármelyik labda, akárkitől kapta) mindig a barátjának rúgja tovább.

Az 1-es játékos barátja a 4-es, a 2-es játékos barátja a 6-os, a 3-as játékos barátja az 1-es, ... , táblázatban:

1	2	3	4	5	6	7	8	9	10	11
4	6	1	3	7	2	11	10	9	5	8

Hányadik sípszó után kerül vissza mindegyik játékoshoz a saját labdája ?

Segítség: Rajzold le körbe a játékosokat és kövesd az egyes labdák útvonalát.

D2. Feladat: /10 pont/

Hányféleképpen lehet 50 azonos csokoládét szétrakni háromféle dobozba: kicsi, közepes és nagy dobozokba? A kicsi dobozokba 2, a közepesekbe 3 és a nagy dobozokba 5 csokoládé fér.

Csak az a lényeg, hogy milyen dobozból hányat töltöttünk meg (lehet, hogy valamelyik dobozból egyet sem). Csak teljesen megtöltött dobozaink lehetnek. Az is lehet, hogy csak egy- vagy kétféle dobozt töltünk meg.

Egy megoldás például: 2*9 + 3*4 + 5*4 = 50
vagyis: 9 db kicsi, 4 db közepes és 4 db nagy doboz van teli.

Segítség: Először rögzítsd, hogy nagy dobozból hányat töltesz meg. Ezután tedd a maradék csokoládékat közepes és kis dobozokba.

D3. Feladat: /10 pont/

PexManó bűvös számát a következőképpen számoljuk ki:

```
1.   A:=3;   B:=7;   C:=-1;
2.   A:=A+1;
3.   For i:=1 to 5 do (A:=A+2;)
4.   D:=B+C;
5.   For i:=1 to 3 do (A:=A+D-i;)
6.   PexBuvos := A+B+C+D ;
```

Mennyi a `PexBuvos` *szám értéke?*

Segítség: Az `"X:=...;"` utasítás hatására a számítógép először kiszámolja az `"..."` kifejezést (képletet), majd ezt az értéket írja be az `"X"` változóba (memóriába).

A `"For i:=1 to y do (...)"` sor esetén `i` értéke `1` -től `y` -ig változik egyesével, és mindegyik `i` érték esetén a zárójelben levő utasítás végrehajtódik.

Készíts táblázatot: a program mindegyik sora után írd be a táblázatba `A,B,C,D` értékét.

A Döntő feladatainak megoldása

D1. Feladat:

Ez majdnem azonos a 4. feladattal. Lássuk, hogyan mennek körbe a labdák:

1	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
4	6	**1**	**3**	**7**	**2**	**11**	**10**	**9**	**5**	**8**

1 => 4 => 3 , 2 => 6 , 5 => 7 => 11 => 8 => 10 , 9

| <= | , | <= | , | <= <= <= | **9**

tehát *lkkt*(3,2,5,1) = 30 miatt a megoldás **30** .
(Részletesebben lásd a 4. feladat megoldásánál.)

D2. Feladat:

Ez majdnem azonos a 8. feladattal. Az összes eset felsorolása:

```
 1)  2*0 +3*0 +5*10= 50
 2)  2*1 +3*1 +5*9 = 50
 3)  2*2 +3*2 +5*8 = 50
 4)  2*5 +3*0 +5*8 = 50
 5)  2*0 +3*5 +5*7 = 50
 6)  2*3 +3*3 +5*7 = 50
 7)  2*6 +3*1 +5*7 = 50
 8)  2*1 +3*6 +5*6 = 50
 9)  2*4 +3*4 +5*6 = 50
10)  2*7 +3*2 +5*6 = 50
11)  2*10+3*0 +5*6 = 50
12)  2*2 +3*7 +5*5 = 50
13)  2*5 +3*5 +5*5 = 50
14)  2*8 +3*3 +5*5 = 50
15)  2*11+3*1 +5*5 = 50
16)  2*0 +3*10+5*4 = 50
17)  2*3 +3*8 +5*4 = 50
18)  2*6 +3*6 +5*4 = 50
19)  2*9 +3*4 +5*4 = 50
20)  2*12+3*2 +5*4 = 50
21)  2*15+3*0 +5*4 = 50
22)  2*1 +3*11+5*3 = 50
23)  2*4 +3*9 +5*3 = 50
24)  2*7 +3*7 +5*3 = 50
25)  2*10+3*5 +5*3 = 50
26)  2*13+3*3 +5*3 = 50
27)  2*16+3*1 +5*3 = 50
28)  2*2 +3*12+5*2 = 50
29)  2*5 +3*10+5*2 = 50
30)  2*8 +3*8 +5*2 = 50
31)  2*11+3*6 +5*2 = 50
32)  2*14+3*4 +5*2 = 50
33)  2*17+3*2 +5*2 = 50
34)  2*20+3*0 +5*2 = 50
35)  2*0 +3*15+5*1 = 50
36)  2*3 +3*13+5*1 = 50
37)  2*6 +3*11+5*1 = 50
38)  2*9 +3*9 +5*1 = 50
39)  2*12+3*7 +5*1 = 50
40)  2*15+3*5 +5*1 = 50
41)  2*18+3*3 +5*1 = 50
42)  2*21+3*1 +5*1 = 50
43)  2*1 +3*16+5*0 = 50
44)  2*4 +3*14+5*0 = 50
45)  2*7 +3*12+5*0 = 50
46)  2*10+3*10+5*0 = 50
47)  2*13+3*8 +5*0 = 50
48)  2*16+3*6 +5*0 = 50
49)  2*19+3*4 +5*0 = 50
50)  2*22+3*2 +5*0 = 50
51)  2*25+3*0 +5*0 = 50 .
```

Tehát a válasz: **51** - féleképpen.

A **rekurzív összefüggés:**

$$a_n = a_{n\text{-}2} + a_{n\text{-}3} + a_{n\text{-}5} - a_{n\text{-}5} - a_{n\text{-}7} - a_{n\text{-}8} + a_{n\text{-}10} \;, \qquad (***)$$

a kezdeti értékek (K.É.P.):

n =	0	1	2	3	4	5	6	7	8	9	10
a_n =	1	0	1	1	1	2	2	2	3	3	4

és a folytatás:

a_{11} = 3+ 3+ 2- 2- 1- 1+ 0 = 4
a_{12} = 4+ 3+ 2- 2- 2- 1+ 1 = 5
. . .
a_{50} = 47 +45 +42 -42 -38 -37 +34 = **51** .

(A magyarázatot lásd 4. feladatnál, (***) -ben nem véletlenül esik ki $a_{n\text{-}5}$.)

A generátorfüggvény:

$$\frac{1}{(1-x^2)(1-x^3)(1-x^5)} = 1 + x^2 + x^3 + x^4 + 2x^5 + 2x^6 + 2x^7 + 3x^8 + 3x^9 + 4x^{10} + 4x^{11} +$$
$$+ 5x^{12} + 5x^{13} + 6x^{14} + 7x^{15} + 7x^{16} + 8x^{17} + 9x^{18} + 9x^{19} + 11x^{20} + 11x^{21} + 12x^{22} + 13x^{23} + 14x^{24}$$
$$+ 15x^{25} + 16x^{26} + 17x^{27} + 18x^{28} + 19x^{29} + 21x^{30} + 21x^{31} + 23x^{32} + 24x^{33} + 25x^{34} + 27x^{35} + 28x^{36}$$
$$+ 29x^{37} + 31x^{38} + 32x^{39} + 34x^{40} + 35x^{41} + 37x^{42} + 38x^{43} + 40x^{44} + 42x^{45} + 43x^{46} + 45x^{47} + 47x^{48}$$
$$+ 48x^{49} + \mathbf{51x^{50}} + 52x^{51} + 54x^{52} + 56x^{53} + 58x^{54} + \ldots$$

D3. Feladat:

		A	B	C	D	D-i	A+D-i
1.	`A:=3;  B:=7;  C:=-1;`	**3**	**7**	**-1**	**0**		
2.	`A:=A+1;`	**4**	**7**	**-1**	**0**		
3.	`For i:=1 to 5 do (A:=A+2;)`	**14**	**7**	**-1**	**0**		
4.	`D:=B+C;`	**14**	**7**	**-1**	**6**		
5.	`For i:=1 to 3 do`					**5**	**19**
	`(A:=A+D-i;)`	**19**	**7**	**-1**	**6**		
	`i:=2`					**4**	**23**
		23	**7**	**-1**	**6**		
	`i:=3`					**3**	**26**
		26	**7**	**-1**	**6**		
6.	`PexBuvos := A+B+C+D ;`	**26**	**+7**	**-1**	**+6 = 38**		

Tehát `PexBuvos` *szám értéke:* = **38** .

PANNON EGYETEM
MŰSZAKI INFORMATIKAI KAR

https://pexpedicio.mik.uni-pannon.hu