



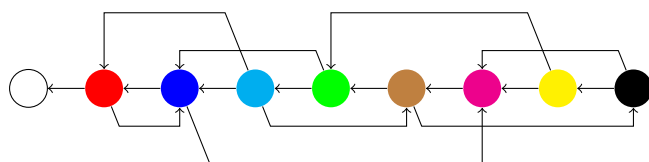
L1 Mazání tabule (řešení)



InterLoS 2022

Tato úloha měla řadu různých strategií a možných řešení.

Při řešení se mohlo hodit zakreslit si, jak je možné barvy překreslovat. Tato informace pomáhala rozhodnout, které barvy zvolit při překreslování konkrétního políčka.



Jednou možnou strategií bylo dostat tabuli nejdříve do stavu, kdy je každé políčko dosažitelné od některého kraje políčky stejné barvy. Takto mohla vypadat tabule po 43 tazích:

Řešení úlohy 81 tahy najdete v souboru [@mazani-tabule-solution.txt](#). Formát je stejný jako formát mezivýsledků zobrazených na webu přes Debug info.

Výsledné heslo: *LAMPION*



L2 Among Los (řešení)



InterLoS 2022

Když si rozepíšete, kdo kdy kde byl, zjistíte, že sídlo výboru (tam byl dort) navštívili a o dortu tvrdili v pořadí:

- Drahoslav, dort tam je
- Cecil, dort tam není
- Bartoloměj, dort tam je
- Cecil, dort tam není
- Emanuel, dort tam není
- Graham, dort tam není
- Frederik, dort tam není

Sus scrofa jsou tedy Cecil a Emanuel a posádka se jednohlasně rozhodla je vyhodit z lodě.

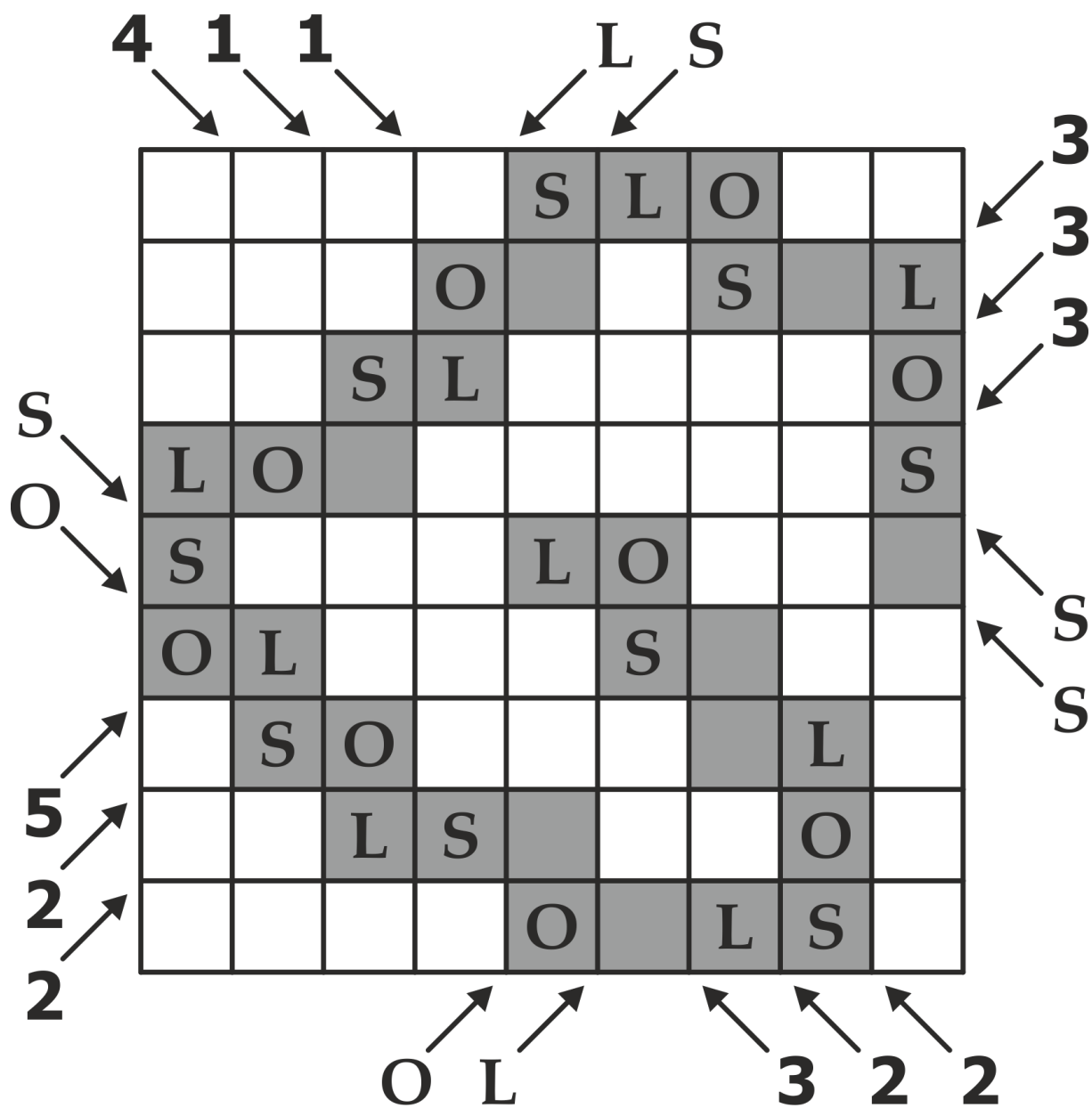
Výsledné heslo: *CE*



L5 LOSÍ stezka (řešení)



InterLoS 2022



Výsledné heslo: *LOSXXLOSLXOXSLOSLOSLOXSLOXSLOSXLOSX*



L6 Losí věže (řešení)



InterLoS 2022

Pro získání hesla je třeba vykonat následující tahy:

1. Přesunout 2 disky z 1. tyče na 3. tyč.
2. Přesunout 3 disky z 1. tyče na 2. tyč.
3. Přesunout 1 disk z 2. tyče na 1. tyč.
4. Přesunout 2 disky z 3. tyče na 2. tyč.
5. Přesunout 3 disky z 1. tyče na 3. tyč.
6. Přesunout 1 disk z 2. tyče na 3. tyč.
7. Přesunout 2 disky z 3. tyče na 1. tyč.
8. Přesunout 3 disky z 2. tyče na 3. tyč.
9. Přesunout 1 disk z 1. tyče na 3. tyč.
10. Přesunout 2 disky z 3. tyče na 1. tyč.
11. Přesunout 3 disky z 1. tyče na 3. tyč.

Výsledné heslo: *LOSIAPOKALYPSAPRICHAZI*



L7 LOSudoku (řešení)



InterLoS 2022

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

Výsledné heslo: 1293867136185269127827974596241993785166945



L8 Slova s L, O, S, I (řešení)



InterLoS 2022



Výsledné heslo: *NVBATEKAJCRKAJHPCMAEPREBN*



L9 Rubikův toroid (řešení)



InterLoS 2022

Řešením byla například následující sekvence tahů:

```
[  
  [[0,2],[1,0]],  
  [[1,0],[0,1]],  
  [[3,2],[0,-1]],  
  [[0,1],[1,0]],  
  [[0,0],[0,1]],  
  [[2,2],[0,-1]],  
  [[0,0],[1,0]],  
  [[3,2],[-1,0]],  
  [[1,2],[0,-1]],  
  [[3,0],[0,1]],  
  [[3,1],[-1,0]],  
  [[0,2],[0,-1]]  
]
```

Formát je stejný jako formát mezivýsledků zobrazených na webu přes Debug info.

Výsledné heslo: *MANDRAGORA*



S1 Cesta kolem světa (řešení)



InterLoS 2022

Návod k sestavení docela jasně říká, jak při řešení postupovat. Nejdříve je nutné objekt vystřihnout a slepit tak, abychom dostali deset čtyřstěnů navzájem spojených hranami, jakéhosi hada. Všimneme si, že hada je možné mnoha způsoby poskládat do tvaru dvacetistěnu (s tím, že polovina dvacetistěnu je vždy prázdná a polovinu zabírá had).

Pokračujeme druhým krokem v návodu. Ten říká, že máme hada složit tak, abychom dostali zeměkouli. Když máme správně složeno, nastává nejdůležitější krok. Je potřeba si všimnout, že nám vznikla cesta procházející přes celý objekt. Tato cesta má celkem 28 políček, jedno z nich je však startovní a druhé koncové. Na zbylých 26 napíšeme anglickou abecedu.



Nastává čas přistoupit k poslednímu kroku. Potřebujeme hada přeskládat tak, abychom dostali všechny tři dílky se žlutým rohem k sobě a všech sedm zelených taky. Když se nám to povede, na objektu se objeví cesta začínající slovem heslo a končící šipkou, jejíž písmenka pouze přečteme.



Výsledné heslo: *TRUS*



S2 Cestopis a zvířata (řešení)



InterLoS 2022

Když si jednotlivé sloky pročítáme, můžeme si všimnout, že jejich text se snaží na něco napovědět. Některé popisy jsou dost přímočaré, takže je poměrně snadné zjistit, že se jedná o české písničky. Pro vyřešení šifry je potřeba většinu písniček identifikovat, u některých možná bude potřeba obrátit se na google.

Když máme písničky určené, můžeme se zamyslet, co s nimi dál. Rozložení slok na stránce podezřele připomíná Braillovo písmo. Jenže které písničky jsou tečky? V tom nám napoví název šifry. Vždyť los na svých cestách hledá zvířata! Zjistíme si tedy, které písničky obsahují v textu zvíře, a braille čteme podle toho.

Chodím po Broadwayi (pes, kočka, prase)	Niagara ()
Tisíc mil (kůň)	Dvě spálený srdce ()
Zpátky do lesů (pes)	Blízko Little Big Hornu (koně)
Holubí dům (holubi)	Okoř ()
Mám malý stan ()	Sbohem láska ()
Červená řeka (hadí kůže)	Rosa na kolejích (pták)
Tulácký ráno (bobřík)	Rodné údolí ()
Řekni, kde ty kytky jsou ()	Sáro (kůň, páv)
Ruty Šuty Arizona (vůl)	Hlídač krav (krávy)

Výsledné heslo: *VUZ*



S3 Druhová (řešení)



InterLoS 2022

Název šifry napovídá, že hledáme nějaké druhy – co druhová jména zvířat? Uspořádání obrázků navíc říká, že tato zvířata žijí jen na jižní polokouli. Teď už je nám jasné, že hledáme tučňáky. Vidíme, že obrázky sedí – tučňák císařský, oslí, brýlový, žlutooký, žlutorohý, kroužkový, královský, novozélandský a skalní.

Heslo už získáme snadno. Obrázek vedle zeměkoule je totiž černý bez, stačí tedy vzít tučňáka bez č, což je tuňák.

Výsledné heslo: *TUNAK*



S4 Milostná poezie (řešení)



InterLoS 2022

Jak může napovědět i název, v básni se vyskytuje spousta slov, která obsahují podřetězec „*los*“. V některých řádcích takové slovo je, v některých ne. Jelikož má každá sloka právě 5 veršů, napadne nás, že se díváme na binárku. Když přiřadíme jednotlivým slokám písmena, získáme mezitajenku „*osma slova*“. Po přečtení osmého slova z každé sloky nám vyjde: „Jak se nazývá druhé největší město naší krásné země“.

Výsledné heslo: *BRNO*



S5 Cizokrajné obrázky (řešení)



InterLoS 2022

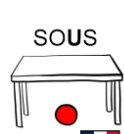
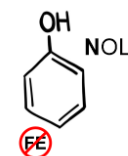
Když správně pojmenujeme obrázky, všimneme si, že každé slovo je vždy složené pouze z písmen L, O, S a jednoho dalšího písmene. Každé písmeno abecedy (LOS nahrazen LÓŠ) se vyskytuje právě v jednom slově. Mohli bychom tedy nalézt substituci a pouze přepsat změť písmen dole. To je skutečně řešení – substituci dává trinárka zakódovaná písmeny L, O, S.



CSS
body {
color: black }



LOS
012



SQL
SELECT *
FROM



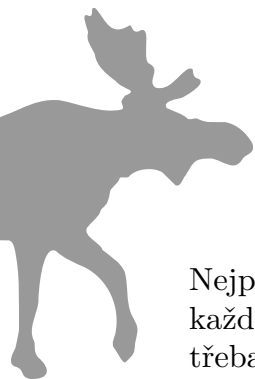
B → OSS → 122 → Q
 C → SS → 22 → H
 D → OLL → 100 → I
 F → OSO → 121 → P
 G → SSO → 221 → Y
 J → OO → 11 → D
 K → OLO → 101 → J
 M → OOS → 112 → N
 N → OL → 10 → C
 V → OOO → 111 → M
 E → OSL → 120 → O
 W → OLS → 102 → K
 X → O → 1 → A
 P → OOL → 110 → L
 A → SLS → 202 → T
 I → SLL → 200 → R
 H → SOO → 211 → V
 L → SO → 21 → G
 Ó → SLO → 201 → S
 Š → OS → 12 → E
 U → SOS → 212 → W
 Q → SL → 20 → F
 R → SSS → 222 → Z
 T → SOL → 210 → U
 Y → SSL → 220 → X
 Z → S → 2 → B



JZOEUCRDNOCSTZHVRUOJUTGSSEO
KREŠTNIJMENÓARCHITEKTAŁÓÓŠE

KŘESTNÍ JMÉNO ARCHITEKTA LOOSE

Výsledné heslo: *ADOLF*



S6 PREKLAD (řešení)



InterLoS 2022

Nejprve je třeba provést překlad českých slov do angličtiny (počet podtržitek pod každým českým slovem odpovídá právě počtu písmen v anglickém překladu). Dále je třeba povšimnout si, že každé anglické slovo má nějaký souměrný prefix nebo suffix. Při překladu (přehnutí) českého slova podle osy souměrnosti této části anglického slova dostaneme zobrazení písmen, jak napovídají symboly v první části šifry.

Získaná zobrazení jsou postupně:

- KARIERA $\{(A, R), (K, I)\}$
- MILIMETR $\{(E, T), (M, R)\}$
- UTOK $\{(T, O), (U, K)\}$
- PRILOHA $\{(R, I), (P, L)\}$
- POLEDNE $\{(L, E), (O, D), (P, N)\}$
- AFERA $\{(E, R), (F, A)\}$
- PRISTEHOVALEC $\{(T, E), (S, H), (I, O), (R, V), (P, A)\}$

Při postupné aplikaci těchto zobrazení na název šifry (uvedený monospace fontem a bez diakritiky, stejně jako slova v šifře) dostáváme heslo.

Výsledné heslo: *VODOVOD*



S7 Teorie multigrafů (řešení)



InterLoS 2022

Z toho, že je multigrafů omezený počet, a z úvodních dvou příkladů, si můžeme všimnout, že multigrafy označují písmenka.

Věty sice rozhodně neplatí pro všechny multigrafy, ale jak říká úvodní text, dají se snadno dokázat pouhým spočtením příkladů. U každé věty tedy spočítáme, kolik písmenek ji splňuje, a tolikáté písmenko z důkazu pak vezmeme.

1. $Q \rightarrow 1 \rightarrow$ Triviální.
2. $A, B, D, E, F, H, K, P, R, T, X, Y \rightarrow 12 \rightarrow$ Silně vám nedoporučujeme...
3. $A, F, G, K, L, O, P, Q, V, X \rightarrow 10 \rightarrow$ Ačkoliv ho probírat...
4. všechna písmena kromě D a O $\rightarrow 24 \rightarrow$ V příloze C. (Určeno jedině pro...
5. $A, O, Q \rightarrow 3 \rightarrow$ Byl ponechán...

Výsledné heslo: *TOPOL*



S8 Kde domov můj (řešení)



InterLoS 2022

Je vidět, že každá kachnička má u sebe svou jmenovku. Podle počtu pípnutí dané kachničky vezmeme ze jména jedno písmeno. V každém záběru tak získáme přečtením písmen jedno slovo. Dostaneme tato slova: Odra, Bečva, (nic), Dehtář, (nic), Labe, Nesyt, Svět. Můžeme si všimnout, že jde o názvy českých řek a rybníků. Pokud interpretujeme řeky jako čárky, rybníky jako tečky a kachny Ignáce a Ivana, jež nepípají, jako oddělovače, dostaneme: ”-./.-..”. Převedením z Morseovy abecedy získáme heslo MED.

Výsledné heslo: *MED*



S9 Na hlavnej ceste (řešení)



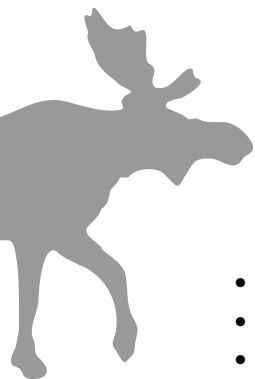
InterLoS 2022

Začneme tým, že obrázky pomenujeme a zamyslíme sa, čo majú spoločné. Prídeme na to, že ku každému sa dá priradiť prídavné meno označujúce štát tak, aby sme dostali známe zaužívané slovné spojenie. Tieto slovné spojenia sú:

- francúzske zemiaky
- veľký čínsky múr
- arabské číslice
- švédsky stôl
- španielska čižma
- zlaté české ručičky
- poľský kríž
- turecký sed
- ruská ruleta
- švajčiarske hodiny
- fínska sauna
- americký sen
- mexická vlna

Ďalej by sme chceli získať písmená tajničky. Na obrázkoch sa nachádzajú rozhádzané značky hlavná cesta v rôznych počtoch, mohli by sa teda využiť ich počty ako indexy. Ak pomocou nich skúsime vyberať písmenká z prídavných mien či pomenovaní obrázkov, nevyjde nám nič rozumné. Preto sa oplatí pýtať, prečo sú to práve hlavné cesty? Čo ešte je hlavné? Ako využiť získané štáty?

Odpoveďou na tieto otázky sú hlavné mestá. Ak povyberáme písmenká z hlavných miest štátov na pozíciách daných počtom značiek, dostaneme:



S9 Na hlavnej ceste (*pokračování*)



InterLoS 2022

- PaříŽ
- PEking
- Rijád
- ŠTokholm
- MaDrid
- PrAha
- Vařšava
- Ankara
- Moskva
- BErn
- HelSinki
- WashingTon
- MexicO City

Vyšlo ŽERT DÁVÁ MĚSTO. Čo s tým? Spravíme to, čo sme robili v celej šifre. Žart sa spája s prídavným menom kanadský, čo nám dáva hlavné mesto Ottawa, ktoré je riešením šifry.

Výsledné heslo: *OTTAWA*



P1 Lámání hesla (*řešení*)



InterLoS 2022

V řešení bylo možné využít takzvaný *timing side-channel attack*: zneužít fakt, že doba, která uplyne, než je heslo zamítnuto, prozrazuje, jak dlouhý prefix sdílí zkoušené heslo s tajným heslem.

Možným řešením bylo tedy postupně budovat správný prefix hesla tím, že ho prodlužujeme o všechna velká písmena abecedy bez diakritiky a jako další znak hesla vybrat to, jehož porovnání trvá nejdéle (nebo pro který dotaz vrátí informaci, že získané heslo je správné).

Pro zajímavost: Pokud jste v řešení při každém průchodu písmeny abecedy vzali první, na jehož odpověď jste čekali déle než vteřinu, pak jste v závislosti na zvoleném pořadí prováděli různé počty požadavků. Pokud jste za pořadí písmen vzali standardní abecední pořadí, bylo třeba provést 123 požadavků. Při pořadí znaků, v jakém jdou na QWERTY klávesnici bylo třeba vyzkoušet 121 možností. Při použití pořadí dle frekvence znaků v angličtině stačilo 100 požadavků a při frekvenci dle češtiny (se zanedbáním interpunkce) jen 96.

Výsledné heslo: *HVEZDARNY*



P4 Kuličkový počítač (*pokračování*)



InterLoS 2022

```
SWAP(0,2);  
start:TEST(1,2);  
STOP();  
ADD(0);  
REMOVE(1,2);  
JUMP(start);
```

n-tý bit v čísle

Pro zjištění hodnoty n -tého bitu si rozdělíme program do dvou fází.

1. V první fázi vydělíme celočíselně zadané číslo dvěma n -krát, tedy provedeme bitový posuv doprava o n bitů.
2. Ve druhé fázi poté zjistíme, zda je výsledné číslo liché či sudé.

Celočíselné dělení dvěma provedeme tak, že si zvolíme pomocný pytlík (například pytlík 2) a pořad dokola opakujeme následující sekvenci příkazů:

1. odeber 2 kuličky z pytlíku 0
2. přidej 1 kuličku do pytlíku 2

tak dlouho, dokud nebude pytlík 0 prázdný. Ještě je potřeba po odebrání první ze 2 kuliček z pytlíku 0 zkontrolovat, zda není pytlík náhodou prázdný a v takovém případě kuličku do pytlíku 2 neházet (jelikož u dělení lichých čísel zaokrouhlujeme vždy dolů). Výsledek dělení pak bude umístěn v pytlíku 2, tak jej už pouze vyměníme s prázdným pytlíkem 0.

Pro zjištění, zda je číslo sudé a nebo liché, můžeme použít následující přístup: nejprve si zvolíme pomocný pytlík, například pytlík 1. Poté opakujeme následující sekvenci příkazů:

1. odeber kuličku z pytlíku 0
2. přidej kuličku do pytlíku 1
3. odeber kuličku z pytlíku 0
4. odeber kuličku z pytlíku 1
5. odeber kuličku z pytlíku 0
6. přidej kuličku do pytlíku 1
7. odeber kuličku z pytlíku 0
8. odeber kuličku z pytlíku 1
9. ...

tak dlouho, dokud nebude pytlík 0 prázdný. Po skončení programu bude v pytlíku 1 uložena kulička právě tehdy, když bylo v pytlíku 0 liché číslo, tedy námi hledaný výsledek. Poslední, co tedy stačí udělat, je prohodit pytlíky 0 a 1.



P4 Kuličkový počítač (*pokračování*)



InterLoS 2022

Program pro hledání n-tého bitu v čísle může tedy vypadat například takto:

```
start:
    TEST(1);
    JUMP(phase2);
bitShiftStart:
    TEST(0);
    JUMP(bitShiftEnd);
    REMOVE(0);
    TEST(0);
    JUMP(bitShiftEnd);
    REMOVE(0);
    ADD(2);
    JUMP(bitShiftStart);
bitShiftEnd:
    SWAP(0,2);
    REMOVE(1);
    JUMP(start);
phase2:
    TEST(0);
    JUMP(end);
    ADD(1);
    REMOVE(0);
    TEST(0);
    JUMP(end);
    REMOVE(1);
    REMOVE(0);
    JUMP(phase2);

end:
    SWAP(0,1);
    STOP();
```



P4 Kuličkový počítač (*pokračování*)



InterLoS 2022

Řazení

V této části úlohy je vstup v pytlících 0 až 4. Založíme si tedy pomocné pytlíky 5 až 9, do kterých si budeme budovat výsledek, přičemž pytlík 5 odpovídá pytlíku 0, pytlík 6 pytlíku 1 atd.

Program bude probíhat iterativně. V každé iteraci provedeme následující kroky:

1. Přerovnáme pytlíky 0 až 4 tak, že prázdné pytlíky budou nazačátku (a ostatní libovolně seřazené)
2. Pro každý neprázdný pytlík 0 až 4 přidáme do odpovídajícího pytlíku 5 až 9 jednu kuličku
3. Odebereme 1 kuličku ze všech pytlíků 0 až 4

Tyto 3 kroky opakujeme tak dlouho, dokud nebudou pytlíky 0 až 4 prázdné. Po skončení této fáze algoritmu bude v pytlících 5 až 9 uložené seřazené pole ze zadání a tedy už jen stačí pytlíky popřehazovat, aby byl výsledek uložen v pytlících 0 až 4.

Nyní už jen stačí dořešit, jak dostat prázdné pytlíky na začátek pole a nezměnit obsah ostatních pytlíků. To uděláme tak, že se podíváme na pytlík 4 a pokud je prázdný, prohodíme jej s pytlíkem 3. Poté uděláme to stejný pro pytlíky 3, 2 a následně 1. Pokud tento postup zopakujeme 4x, budemem mít vždy prázdné pytlíky na začátku sekvence a zároveň, jelikož pouze vyměňujeme pytlíky, a jinak nemanipulujeme s jejich obsahem, tak se nám žádné kuličky neztratí či nepromíchají.

Program pro řazení pole délky 5 může tedy vypadat například takto:

```
ADD(10);
start:
  ADD(11);
  ADD(11);
  ADD(11);
  ADD(11);
  swStart:
    TEST(11);
    JUMP(swEnd);
    TEST(1);
    SWAP(0,1);
    TEST(2);
    SWAP(1,2);
    TEST(3);
    SWAP(2,3);
    TEST(4);
```



P4 Kuličkový počítač (*pokračování*)



InterLoS 2022

```
    SWAP(3,4);
    REMOVE(11);
    JUMP(swStart);
swEnd:

TEST(0);TEST(10);
ADD(5);

TEST(1);TEST(10);
ADD(6);

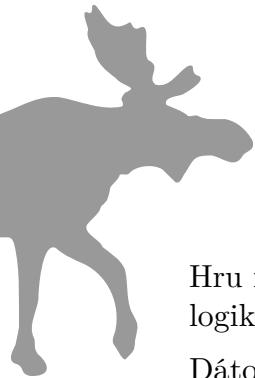
TEST(2);TEST(10);
ADD(7);

TEST(3);TEST(10);
ADD(8);

TEST(4);TEST(10);
ADD(9);

REMOVE(0,1,2,3,4);
TEST(0,1,2,3,4);
JUMP(end);
JUMP(start);
end:
REMOVE(10);
SWAP(0,5);
SWAP(1,6);
SWAP(2,7);
SWAP(3,8);
SWAP(4,9);
STOP();
```

Výsledné heslo: *LOPATASEKTAKACHNA*



P5 Kartová hra (řešení)



InterLoS 2022

Hru našich piatich losov stačí priamočiaro odsimulovať tým, že naprogramujeme všetku logiku hry popísanú v zadaní.

Dátová štruktúra, ktorá nám to umožní riešiť pohodlne, je queue (fronta) alebo deque (obojstranná fronta). Tú použijeme ako na ruky hráčov, tak aj na balíček na stole, pretože vždy pridávame nové karty a odoberáme najstaršie. Na odhadzovací a doberací balíček môžeme použiť tú istú frontu, pretože staré karty odhadzovacieho balíčka sa objavujú naspodu doberacieho (ako najnovšie) a je jedno, kedy by sa karty presunuli z odhadzovacieho do doberacieho.

Výsledný balíček vyzerá takto:

```
zelen 9 (H)
zelen 10 (E)
zelen 7 (S)
zelen dolnik (L)
gula dolnik (O)
gula 8 (J)
gula kral (E)
srdce kral (I)
srdce dolnik (N)
zalud dolnik (F)
zalud 7 (R)
zalud 10 (A)
zalud 9 (S)
gula 9 (T)
gula 10 (R)
gula hornik (U)
zelen hornik (K)
srdce hornik (T)
srdce 10 (U)
srdce 8 (R)
srdce eso (A)
```

Prečítaním heslotvorných písmeniiek sa dozvieme, že HESLOJEINFRASTRUKTURA.

Výsledné heslo: *INFRASTRUKTURA*



P6 Megabludisko (řešení)



InterLoS 2022

Keby sme sa pokúsili vygenerovať celé bludisko, už len jeho vytvorenie by bolo príliš pomalé. Navyše, ešte skôr by sme došli k problému, že nám pravdepodobne nebude stačiť pamäť bežného počítača. Musíme preto nejako využiť štruktúru bludiska – teda to, že je zložené z rovnakých dlaždíc. Navyše dlaždice majú na obvoде veľké množstvo stien, čo znamená, že počet prechodov medzi dlaždicami je pomerne malý.

V každej dlaždici sa dá pohybovať rovnako. Pohyb vždy začneme na okraji dlaždice, môžeme si teda na začiatku predpočítať vzdialenosť z každého okrajového políčka dlaždice na všetky ostatné políčka dlaždice. Na to môžeme využiť napríklad prehľadávanie do šírky (BFS) z každého krajného políčka, ktoré nie je stena, samostatne.

Pre jednoduchšiu implementáciu pridáme ku každej dlaždici ešte na pravý okraj ľavý stĺpec a na spodný okraj horný riadok, pretože takto budeme vedieť jednoducho nadpájať cesty cez jednotlivé dlaždice za seba. Keby sme to nespravili, pri nadpájaní ciest z rôznych dlaždíc by sme ešte museli vkladať jeden krok, ktorý musíme prejsť z jednej dlaždice na druhú.

Ďalej si vytvoríme graf, ktorého vrcholy budú políčka, ktoré ležia v hornom riadku alebo ľavom stĺpci niektorej dlaždice, a cieľové políčko v pravom dolnom rohu. Pre každú dlaždicu pridáme do grafu hrany medzi voľnými políčkami na okraji, ktorých dĺžky zistíme z predpočítaných vzdialeností. Nesmieme zabudnúť pridať všetky hrany medzi všetkými stranami dlaždice, dokopy je šesť kombinácií (ľavá – horná, ľavá – pravá, ľavá – dolná, horná – pravá, horná – dolná a pravá – dolná), pričom dolná strana je zhodná s hornou stranou dlaždice o riadok nižšie a pravá je zhodná s ľavou stranou dlaždice o stĺpec vpravo. Napokon ešte pridáme hrany zo štartovného políčka a cieľového políčka.

Ak by sme skúšali všetky dvojice políčok, ktoré ležia na okraji jednej dlaždice, program by bol príliš pomalý. Ak máme políčko, ktoré je stena, preskočíme ho hneď a negenerujeme k nemu všetky možnosti ostatných políčok.

Získali sme ohodnotený graf a zaujíma nás dĺžka najkratšej cesty zo štartovného do cieľového políčka, takže môžeme použiť Dijkstrov algoritmus.

Výsledné heslo: *40239*

Vzorový kód v Pythone si môžete pozrieť tu: [@megaBludisko-solution.py](https://github.com/megaBludisko-solution.py).



P7 Moose Airlines (řešení)



InterLoS 2022

V úlohe máme zadané letiská a sieť letov medzi nimi a chceme nájsť najlacnejšiu cestu spĺňajúcu dané podmienky. Dôležitou vlastnosťou, ktorú lety majú, je, že skorší let použijeme vždy skôr ako neskorší, ak ich použijeme oba. To je významná odlišnosť od prehládavania bežného grafu, kde keď máme dve hrany, môžeme ich použiť v ľubovoľnom poradí.

Lety budeme teda spracúvať podľa času. V riešení sme zvolili usporiadanie podľa času pristátia, pretože počas hlavného cyklu spracúvajúceho lety nám platí, že si pre každé letisko udržujeme cenu najlacnejšej cesty doň, ktorá končí najneskôr v čas pristátia aktuálneho letu. Ak v ten istý čas pristáva viac letov, tak z tejto časovej známky máme spracovanú len časť letov, ale nakoľko každý let trvá nenulový čas, tak to nevadí.

Ako sme už naznačili, budeme si chcieť pre každé letisko udržiavať cenu najlacnejšej cesty doň pomocou doteraz spracovaných letov. Pozrime sa na to, ako túto hodnotu aktualizovať pri spracúvaní ďalšieho letu z letiska A na letisko B , ktorý vyráža v čase t_0 , pristáva v čase t_1 a má cenu c . Zrejme sa môže zmeniť len cena cesty na letisko B a zmení sa, len ak tento let využijeme. Ak ho využijeme, musíme ho nadpojiť na cestu na letisko A , ktorá skončí najneskôr v čase t_0 . (Ak let vyráža z Brna, napojíme ho na cestu ceny 0, ktorá v ňom začína aj končí v čase 0).

Tu vidíme, že si potrebujeme pamätať históriu najlacnejších ciest. Nemôžeme si dovoliť počítať najlacnejšiu cestu pre úplne každú časovú známku (časovo ani pamäťovo). Bude nám však platiť, že v neskorších časoch je cesta lacnejšia alebo rovnako drahá (ak v čase t vieme byť niekde za x peňazí, v ľubovoľnom neskoršom čase tam tiež vieme byť za najviac x peňazí použitím tejto cesty). Stačí nám teda ukladať si nové najlepšie ceny pre časy, kedy sa menia. Tieto ceny sú navyše klesajúce s rastúcim časom.

Ak nás teraz zaujíma najlacnejšia cesta v čase t_0 , v takto uloženej histórii vieme binárne vyhládať, kam by spadol čas t_0 a podľa toho určiť cenu c_A . Môže sa stať, že sa v čase t_0 na letisko A nevieme dostať, v takom prípade práve spracúvaný let nevieme použiť. V opačnom prípade máme možnosť prísť na letisko B v čase t_1 za $c_A + c$ peňazí. Ak je táto cena väčšia alebo rovná ako aktuálne najnižšia cena do B (posledná v histórii), nemá zmysel let využiť. Ak je nižšia, zapíšeme si k letisku B túto cenu s časom t_1 .

Ostáva splniť podmienky cesty. Nutnosť návštevy Aucklandu a Honolulu vyriešime tak, že každé letisko nahradíme štyrmi jeho kópiami, pre každú možnosť, ktoré z týchto letísk sme už navštívili. Každý let na vstupe tak bude vychádzať zo štyroch kópií letiska a bude viesť do kópie s rovnakým stavom navštívenia. Ak však ide o let do Honolulu, resp. Aucklandu, bude viesť do kópie, v ktorej už toto letisko je navštívené.



P7 Moose Airlines (pokračování)



InterLoS 2022

Ďalšou podmienkou je, že v Honolulu musíme stráviť aspoň jeden deň (1440 minút). To vyriešime tak, že letom, ktoré nastavujú príznak, že Honolulu už je navštívené, pripočítame ešte pred zoradovaním letov podľa priletu 1440 minút k dĺžke. To, že musíme stráviť v Honolulu 1440 minút efektívne znamená, že odísť môžeme o 1440 minút neskôr, takže akoby sme celý čas boli v lietadle na ceste tam. Odpočítavať tento čas od letov z Honolulu by fungovalo analogicky tiež.

Výsledné heslo: 77967

Vzorové riešenie v Pythone nájdete tu: [@airlines-solution.py](https://github.com/interlos/airlines-solution.py).



P8 Kríky na priamke (řešení)



InterLoS 2022

Priamka je jednoznačne určená dvomi bodmi. Optimálna priamka, na ktorej rastie najviac bobúľ, bude určite prechádzať aspoň cez dva kríky.

Na základe týchto pozorovaní už vieme vytvoriť prvé (ale príliš pomalé) riešenie úlohy: pre každú dvojicu kríkov spočítame všetky bobule na kríkoch, ktoré ležia na nimi určenej priamke, a vezmeme maximum cez všetky priamky (dvojice kríkov).

Efektívnejšie riešenie vieme založiť na myšlienke, že keď už máme vybrané dva kríky A a B , môžeme si ich zapamätať k priamke, ktorú určujú, a nemusíme prechádzať všetky ostatné kríky. Ak je nejaký iný krík C na tej istej priamke, pri prechádzaní všetkých dvojíc kríkov časom určite narazíme na dvojicu (A, C) (a takisto aj (B, C)), ktorá nám dá tú istú priamku, a tak si na ňu krík C pripíšeme.

Potrebuje však spôsob, ako jednoznačne identifikovať priamku tak, aby sme (ak body A , B , C ležia na jednej priamke) pre dvojicu (A, B) a dvojicu (A, C) , a každú inú dvojicu bodov na tejto priamke, dostali tú istú reprezentáciu a vedeli si k nej zapamätať všetky kríky, ktoré na nej ležia.

Je viacero možností, ako niečo také dosiahnuť. Vo vzorovom riešení sme zvolili možnosť, že si pamätáme sklon priamky a y -ovú súradnicu bodu, ktorý má x -ovú súradnicu nulovú. Zvislé priamky pritom budeme uvažovať samostatne, pretože ich sklon je nekonečno a na $x = 0$ majú nula alebo nekonečno bodov, nie práve jeden.

Riešenie teda bude vyzerat' tak, že postupne pre každú dvojicu kríkov (A, B) spočítame jednoznačnú (kanonickú) reprezentáciu nimi určenej priamky a zapamätáme si k nej oba tieto kríky. Kríky na jednotlivých priamkach si budeme udržiavať v množine (`set` v Pythone), pretože každý krík si chceme ku každej priamke držať iba raz. Ak by sme nefiltrovali duplikáty, každý krík by sa na priamke objavil toľkokrát, koľko iných kríkov leží na danej priamke.



P8 Kríky na priamke (pokračování)



InterLoS 2022

Následne prejdeme všetky priamky, ktoré sa na vstupe objavili, pre každú spočítame počet bobúl a vypíšeme najväčší z nich.

Na optimálnej priamke ležia tieto kríky:

```
135 1540 14
915 505 19
655 850 10
-21 1747 11
1019 367 9
447 1126 18
967 436 20
1331 -47 14
1279 22 20
811 643 19
-177 1954 13
```

Výsledné heslo: *167*

Kód na riešenie úlohy v Pythone môže vyzerat' takto: [@line-solution.py](#).



P9 Loscilátory (řešení)



InterLoS 2022

Řešení nebylo komplikované. Stačilo hodnoty loscilátorů modulovat ve správném pořadí a zkontrolovat, kdy jsou všechny loscilátory na hodnotě 0.

Vzorové řešení je v souboru [@loscillator-solution.py](#).

Výsledné heslo: *88088*