

Soutěž dětí a mládeže v programování

Obvodní kolo Prahy 1 a 2, rok 2025

Autoři: Adam Benda, Martin Horský, David Šertler; verze 1.0, 2025-02-23

Každá úloha má koeficient obtížnosti, kterým se násobí hodnocení vašeho řešení. Odevzdejte nám prosím řešení přes odevzdávátko (<https://soutez.lnker.xyz/>) jako komprimovanou složku zip. Odevzdat jednu úlohu můžete vícekrát - nejlépe po dosažení nějakého stupně funkčnosti. Hodnotíme verzi úlohy, kterou jste odevzdali jako poslední. Kromě splnění zadané funkčnosti se hodnotí přehlednost vašeho kódu (komentáře, popisující názvy proměnných) a efektivita vašeho řešení.

Při soutěži je zakázána komunikace (přítel na telefonu). Vyhledávač a Internetové zdroje používat můžete, stejně jako generátory kódu. Zkopírovat si několik řádků kódu ze stackoverflow (uvedte zdroj) nebo použít knihovni funkci řešící dílčí část problému je v pořádku. V případě pochybností konzultujte porotce.

Používání knihoven je povoleno, jelikož mnoho programovacích jazyků má relativně omezenou standardní knihovnu (např. grafické zobrazování v okýnkách). Nepoužívejte ale knihovny, které víceméně vyřeší úlohu za vás.

Smíte používat LLM chatboty. Pracujte s nimi ale opatrně. Obecně je dobré se nespoléhat vždy na výstup LLM. Tyto AI mohou vkládat do svého výstupu chyby (AI halucinace). Měli byste z nich používat pouze kód, který sami chápete.

Testovací vstupy k úlohám naleznete na <https://soutez.lnker.xyz/res/>.

Po soutěži nám prosím vyplňte anketu na <https://lnker.xyz/anketa>.

Úloha 1: Zákazy (koef. 2)

Vytvořte program, jenž zvládne vygenerovat následující dopravní značky:

- Zákaz vjezdu všech vozidel v obou směrech (klasický bílý kruh s červeným okrajem)
- Zákaz vjezdu traktorů (uvnitř bílého kruhu je traktor)
- Zákaz vjezdu motocyklů (uvnitř bílého kruhu je motocykl)
- Libovolný další zákaz vjezdu kde na vstupu dostanete piktogram zakazované třídy vozidel
- Zákaz vjezdu všech motorových vozidel (kruh je rozdělen na dvě poloviny horizontální červenou čarou, v horní polovině obsahuje piktogram motorového vozidla, ve spodní motocyklu)

- Zákaz vjezdu vyznačených vozidel (dělený na třetiny, v každé z třetin může být jiný piktogram zakazované třídy vozidel)

Umožněte uživateli rozumně vybrat kterou značku chce. Piktogramy můžete načítat z adresáře - ale uživatel musí být schopen je vybrat ze seznamu a případně do adresáře přihrát. Piktogramy najdete v testovacích datech k soutěži.

Značku buď zobrazte uživateli a/nebo uložte obrázek do souboru.



Hodnocení:

- Každý z těchto pěti příkladů správně vygenerovaný - 15%
- Přehledné uživatelské rozhraní - 10%
- Možnost přihrát další obrázky - 5%
- Přehlednost, dokumentace - 10%

Úloha 2: Počet slov (koef. 2)

Často se vám hodí znát počet slov v psaném textu. Minimálně abyste věděli jestli musíte pokračovat v psaní referátu. Tak si pro to napíšeme program. A když už jsme u toho, tak budeme počítat i další věci, jako počet řádek a počet znaků.

Napište program, který na vstupu dostane text a vypíše na výstup počet znaků, slov a řádek. Znak je jeden samostatný symbol. Pokud vstup obsahuje diakritické znaky, jednoduché smajlíky atd., bude vstup v kódování UTF-8 a každý takový znak se počítá jako **jeden znak**. Složené smajlíky, jako vlajky nebo smajlíci se specifickou barvou pleti (🇧🇪, 🇨🇪 a 🍷), můžete počítat jako několik znaků. Pokud ale chcete, můžete zkusit implementovat i rozlišování složených znaků. Slova jsou libovolné sekvence tisknutelných znaků, oddělené alespoň jedním bílým znakem (mezera, tabulátor, nová řádka...). A řádky jsou sekvence znaků oddělené nějakým děličem řádků (LF nebo CRLF).

Vstup můžete brát ze standardního vstupu, ze souboru nebo nějak interaktivně. Pokud budete brát vstup ze souboru, musí váš program mít možnost uživatelsky zadat cestu ke vstupnímu souboru. Konkrétní implementace je na vás.

Výstup programu budou tři čísla. Jestli je vypíšete na standardní výstup, do souboru nebo grafického rozhraní je na vás.

Vaše řešení by nemělo využívat program wc či obdobné nástroje které by problém vyřešily za vás.

Hodnocení:

- Vstup lze zadat uživatelsky (ne fixní soubor) - 5%
- Správně udává počet znaků v ASCII - 10%
- Správně udává počet znaků v UTF-8 - 20%
- Správně udává počet slov v ASCII - 10%
- Správně udává počet slov v UTF-8 - 20%
- Správně udává počet řádek v ASCII - 10%
- Správně udává počet řádek v UTF-8 - 15%
- Přehlednost a efektivita - 10%
- Bonus: počítá složené znaky (🇨🇪) jako jeden - 20%

Úloha 3: Zmatené Šachy (koef. 3)

Ajaj! Váš oblíbený pěšec onemocněl chorobou Metamorphosis Acuta, a vy přitom nutně potřebujete dojít na cílové políčko.

Na prázdné šachovnici je zadáno výchozí a cílové políčko. Na výchozím políčku se na začátku nachází figurka, která mění svoje schopnosti podle rozpisu. Například rozpis "střelec, vez" = v prvním tahu tahne jako střelec, v druhém jako věž, ve třetím opět jako střelec, pak jako věž, a tak dále. Implementujte pohyb:

- Věž (vez) - může tahnout v horizontální linii (např a3->h3) nebo ve vertikální (a3->a8)
- Střelec (střelec) - může tahnout úhlopříčně (šikmo do čtyř směrů)
- Jezdec (jezdec) - Tahne o +2 nebo -2 v jedné souřadnici a o +1 nebo -1 ve druhé - výsledkem je pohyb tvaru písmena L
- Král (kral) - Tahne po vertikále, horizontále či diagonále (tedy do osmi směrů), ale pouze o jedno políčko

	a	b	c	d	e	f	g	h	
8									8
7									7
6									6
5									5
4									4
3									3
2									2
1									1
	a	b	c	d	e	f	g	h	

Vstup máte k dispozici v textové podobě nebo ve formátu JSON. Který formát vstupu bude implementovat je na vás, ale uveďte to poté v popisu řešení.

Výstupem programu bude počet a seznam tahů, kterými se figurka nejrychleji dostane ze svojí výchozí pozice na cílovou. V případě více možných nejkratších řešení stačí uvést jedno. Figurka se musí v rámci svého tahu posunout - nelze tah ignorovat a zůstat na původním místě.

Hodnocení:

- Funkční vyhledání nejkratší cesty - 30%
- Backtracking - vypsání nejkratší cesty - 20%
- Všechny figurky mají správně zpracované tahy (5% * 4)
- Přepínání figurek funguje dle zadání - 20%
- Přehlednost a efektivita - 10%

Úloha 4: Logik přes síť (koef. 4)

Hra Logik je deduktivní hra pro dva hráče. Jeden hráč si vymyslí nějakou sekvenci 4 barev z 6 možných barev. Barvy se mohou v sekvenci opakovat. Druhý hráč zkusí uhádnout sekvenci. Když se trefí, hra končí. Když ne, první hráč mu prozradí kolik barev trefil na správné pozici a kolik trefil v sekvenci, ale na špatné pozici. Druhý hráč poté znovu hádá. Tohle se opakuje, dokud druhý hráč neuhádne celou sekvenci. Možné barvy jsou červená, modrá, žlutá, zelená, černá a bílá.

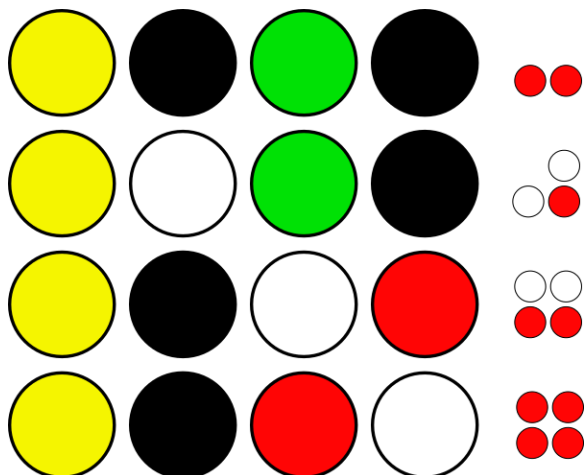
Průběh hry může být například:

Žlutá, černá, zelená, černá — 2 na správném místě, 0 na špatném místě

Žlutá, bílá, zelená, černá — 1 na správném místě, 2 na špatném místě

Žlutá, černá, bílá, červená — 2 na správném místě, 2 na špatném místě

Žlutá, černá, červená, bílá — Správně



Vy budete tuto hru hrát proti počítači. Na serveru Inker.xyz na TCP portech 1234 a 2345 běží program, který s vámi bude tuto hru hrát. Ten na portu 2345 ale funguje trochu obecněji, než jaká jsou původní pravidla hry. Na portu 1234 máte k dispozici standardní pravidla. Sekvence má vždy 4 pozice a možné barvy jsou výše uvedené barvy v angličtině (red, green, blue, yellow, black, white). Ale na portu 2345 je náhodně dlouhá sekvence dlouhá 2-6 sloty, počet barev ze kterých se sekvence může skládat je 4-8 náhodně vybraných ze seznamu pár set.

Komunikace se serverem je textová, probíhá přes TCP a každá zpráva je jeden řádek ukončený znakem LF (ASCII hodnota 10). Když se připojíte, server vám pošle slovo "Welcome", poté délku hádané sekvence a poté barvy ze kterých se sekvence skládá. Dále čeká na zprávu od klienta a po přijetí vyhodnotí pokus a odpoví správný počet na správné pozici a správný počet na špatné pozici. Když se trefíte, server odpoví "Correct" a ukončí spojení.

Výše uvedený příklad by vypadal následovně.

SERVER	CLIENT
Welcome	
4	
red,green,blue,yellow,black,white	
	yellow,black,green,black
2,0	
	yellow,white,green,black
1,2	
	yellow,black,white,red
2,2	
	yellow,black,red,white
Correct	

Pokud kdykoliv pošlete neplatnou zprávu, server vám pošle nějakou chybovou zprávu a ukončí spojení. Za chybu se nepovažují bílé znaky kromě nové řádky (mezera, tabulátor, CR) mezi slovy.

Vaším úkolem je napsat program, který se připojí na náš server a zahraje jednu hru. Váš program musí mít nějaký způsob jak uživatelsky zadat adresu a port serveru. Jak přesně je na vás - argumenty programu, konfigurační soubor, interaktivně... V průběhu by měl váš program také vypisovat informace o pokusech na uhádnutí a odpovědích serveru.

Váš program by měl uhodnout sekvenci po co nejméně pokusech. Optimální řešení originální hry může garantovaně najít řešení do 5 pokusů. Zkuste si nejprve naprogramovat řešení vůči serveru na portu 1234 a až poté zkuste server na portu 2345. Oba servery komunikují totožně, tedy i server na portu 1234 posílá úvodní zprávu s délkou sekvence a možnými barvami.

Váš program může použít pro výpočet více vláken a bude testován s max 10 vlákny. Jeden průběh základní hry by měl trvat nanejvýš 20 sekund.

Jelikož komunikace se serverem je čistě textová, můžete si ji vyzkoušet za pomoci programu telnet nebo ji sledovat programem wireshark.

Hodnocení:

- Program správně komunikuje se serverem a správně zpracovává zprávy serveru - 5%
- Vypisuje informace o komunikaci uživateli - 5%
- Adresa serveru a port jsou nastavitelné - 5%
- Program dokáže vyřešit hru serveru na portu 1234 - 10%
- Program dokáže vyřešit hru serveru na portu 1234, vždy do 20 pokusů - 10%
- Program dokáže vyřešit hru serveru na portu 1234, vždy do 6 pokusů - 30%
- Program dokáže hrát se serverem na portu 2345- 20%
- Přehlednost a efektivita - 15%