Soutěž dětí a mládeže v programování

Obvodní kolo Prahy 1 a 2, rok 2024

Autoři: Adam Benda, Martin Horský, David Šertler; verze 1.1, 2024-02-23

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 ALE: cílem je zjistit vaši schopnost algoritmizace. Zkopírovat si několik řádků kódu ze stackoverflow (uveďte zdroj) nebo použít knihovní funkci řešící dílčí část problému je v pořádku. Zkopírovat odněkud hotové řešení celé úlohy není přípustné. 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.

Současně nepoužívejte AI generátory stylu ChatGPT, GitHub Copilot a podobné. “Chytré našeptávače” nezakazujeme, ale ať vám to negeneruje víc jak pár řádek na myšlenku.

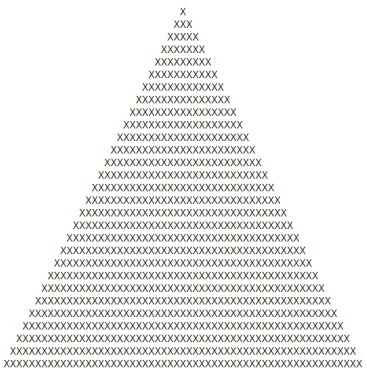
Některé z úloh používají vstupní data pro definici instance problému. Tyto vstupy jsme připravili v několika formátech (hlavně text, JSON a YAML). Pokud je to určeno v zadání úlohy, máte také na výběr, jestli budete vstup načítat ze souboru, standardního vstupu nebo v grafickém rozhraní. Tato rozhodnutí jsou na vás, ale pokaždé nám to napište do přiloženého README.

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

Po souteži nám prosím vyplňte anketu na https://lnker.xyz/anketa.

Úloha 1: Pyramida (koef. 1)

Do standardního textového výstupu (nebo textové oblasti s nastaveným Monospace fontem) vygenerujte pyramidu znaků. První řádek má jeden znak, každé další patro o dva znaky víc než to předchozí. Pyramida je centrovaná - její špička je přímo nad středem základny.
Umožněte uživateli zvolit, který znak se použije a počet pater (řádků) pyramidy.



Úloha 2: Inventář plyšáků (koef. 2,5)

Vytvořte program na evidenci sbírky plyšových zvířátek.

Každé zvířátko má svoje **jméno** (např. Rex, Elsa, Flufík), **druh** (např. žirafa, slon, osmák degu, atp.) a **barvu** (červená, zelená, fialově puntíkatá, atp.). Neomezujte uživatele nabídkou možností, umožněte všechny údaje zadat textově libovolně.

Do inventáře umožněte uživateli interaktivně zadávat (např. formulářem). Můžete udělat i dávkové načtení (ze souboru) pro účely testování, ale nejsou za to body navíc.

Vytvořte program tak, aby vaše evidence vydržela po vypnutí programu / restartu počítače - například zápisem do a načtením ze souboru, databáze, či jiného perzistentního úložiště (JS local storage).

Umožněte hledat v databázi zadáním **jména** - vypište plyšákovy vlastnosti, nebo informaci že nebyl nalezen. Vaše sbírka je velká, dbejte tedy na efektivitu tohoto vyhledání.

Hodnocení:

- Zadávání plyšáků do databáze a jejich zobrazení - 30%
- Stálost dat po restartu programu - 30%
- Hledání v databázi podle jména (včetně efektivity) - 30%
- Přehlednost řešení - 10%

Úloha 3: Já tě vidím (koef. 3)

Nebylo by hezké mít na počítači společníka co neustále dohlíží na naši práci?

Chceme mít na počítači oči, které sledují pohyb kurzoru. Mějme je třeba v rohu a kdykoliv se pohne kurzor, pohnou se i zorničky. Pro jednoduchost budeme mít pro program jedno okno s nějakým pozadím a někde v tomto okně i oči. A kdykoliv je kurzor v tomto okně, tak se na něj oči dívají.

Velikost okna, pozici očí a barvu/obrázek pozadí by měla být nastavitelná. Jestli to bude přes argumenty programu, nebo ze souboru je na vás. Jak toto naimplementujete ale zdokumentujte.

Pokud vás nenapadá jak snímat pozici kurzoru nebo animovat, udělejte alespoň program který vykreslí jeden snímek. Tehdy by váš program dostal navíc pozici na kterou se mají oči koukat. Vytvořený snímek uložte do souboru.

Příklad referenčního řešení viz: https://youtu.be/VQnUw7ksss8

Hodnocení:

- Program vykreslí oči - 20%

- Nastavitelnost umístění očí, barvy, velikosti atd… - 30%
- Oči jsou animované - 40%
- Přehlednost, efektivita řešení - 10%

Úloha 4: Pošta (koef. 3)

Státní pošta v blíže nespecifikované zemi provádí škrty v rozpočtu. Nabízí se třeba utrácet méně za benzín, když v nějaký den, nepotřebují doručit zásilky po celém městě.

Když pošťák jezdí s poštou, má za úkol doručit balíky a dopisy na konkrétní adresy. Často doručuje skoro všude a tak se mu vyplatí prostě projet celou čtvrť. Některé dny je ale zásilek málo. Takže je zbytečné projíždět okolo každého domu. Lepší by bylo, kdyby jel jenom k domům, pro které má zásilku. A také by bylo vhodné při tom ujet co nejkratší vzdálenost.

Váš program dostane mapu města v podobě mřížky s vyznačenými místy kde pošťák začíná (a i končí) a místy kam doručuje. Výstupem vašeho programu bude délka nejkratší okružní cesty a mapa s vyznačenou cestou.

Mapa má podobu mřížky, kde každý znak označuje jednu souřadnici. Znaky mohou být:

- # - zastavěno
- . - cesta
- S - depo (začátek/konec)
- x - adresát

Vstupní a výstupní mapa může vypadat například takto:

```
                                Total distance is: 60
#######..x....######            #######xxxxxx######
#######.#####.######            #######x#####x######
S...................            Sxxxxxxx.....x......
####.#####.##.#####.            ####x#####.##x#####.
####.#####.##.#####.            ####x#####.##x#####.
####.......##x#####.            ####x......##x#####.
####.#####.##.#####.            ####x#####.##x#####.
##x..#####.##......x            ##xxx#####.##xxxxxxx
####.#####.####.####            ####x#####.####x####
####............####            ####xxxxxxxxxxxx####
```

Vstup máte k dispozici buď textově (jak je znázorněno výše) nebo v JSON a YAML v řádkovém uspořádání (pole polí, kde vnitřní pole reprezentují řádky). Můžete předpokládat platný vstup. Nemusíte kontrolovat zda jsou všechny řádky stejně dlouhé nebo zda má mapa právě jedno depo. Také se můžete spolehnout, že všechny silnice (a tedy i adresáti) jsou dosažitelní z depa.

Výstup vašeho programu může být na standardní výstup, do souboru nebo v grafickém rozhraní.

Pro odrážky hodnocení na nejkratší cestu platí, že váš program by měl vždy vrátit nejkratší cestu, případně téměř vždy za sražený počet bodů. Nikoliv ale proporčně. Pokud váš program najde nejkratší cestu pouze v půlce případů, neznamená to nárok na půlku bodů.

Hodnocení:

- Najde cestu za alespoň jedním adresátem a zpět - 5%
- Najde nejkratší cestu za alespoň jedním adresátem a zpět - 10%
- Najde cestu za všemi adresáty a zpět - 15%
- Najde nejkratší cestu za všemi adresáty a zpět - 50%
- Přehlednost a efektivita - 20%