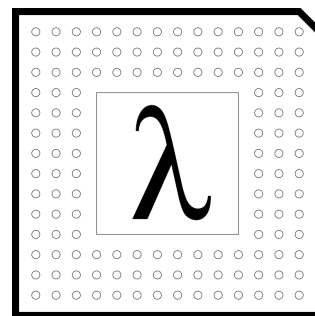


Úloha č. 2

Podivná komunikace II



Odpověz Sfinze!

8 b

Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

Soutěž vyhrána! Jak už bývá zvykem, bylo to o špičku rakety, ale v závěrečném čase 31 minut a 42 sekund tvá posádka předčila stříbro beroucí tým z VŠE. Po poslední vybrané zatáčce, ve které jste se málem srazili s raketou ČZU, všichni popadáte dech a plácáte se po zádech. Tak přece jenom letíme do vesmíru!

Avšak raketa se sama s posádkou se stále ne a ne odlepit od země. Samotné instrukce totiž k letu nestačí, posádka je nezládá do terminálu vypisovat tak rychle, jak by bylo potřeba. Předkládáš tuto poměrně závažnou informaci při jednání představenstva týmu. Obavy jsou zcela na místě, vedoucí týmu totiž o žádném zesložítování čehokoli nechce ani slyšet.

Po dvacetiminutovém přesvědčování se nakonec uvolil ke změně. „Jestli nám ale něco spadne na hlavu,“ hřímal, „za bezpečí posádky zodpovídáte vy. Já o tom nechci slyšet ani slovo, jestli nám někdo na tý palubě odejde k pánu.“

Tvým úkolem bude za pomoci instrukcí vymyšlených v minulém kole napsat program. Nakonec se jich sešlo tolik, že v jazyce nejsou ani zdaleka všechny. Jsou tam ale ty nejzajímavější a nejužitečnější z nich. A rozhodně budou stačit k tomu, abys zvládl napsat jakýkoliv program si jenom usmyslíš.

Interpreter je k dispozici na <https://github.com/fiks-org/fikslang-interpreter>. Doporučím si ho stáhnout a vyvíjet svůj program pomocí něj. Seznam instrukcí najdeš na githubu.¹

Popis stroje

Pracujeme se se stejným strojem jako minule. Pro detailnější připomenutí se podívej na zadání minulé úlohy.

Stručně, jedná se o stack machine, který má k dispozici dvě místa pro ukládání dat. Stack a paměť. Paměť umožňuje přístupy přes index, a to jak pro zápis, tak pro čtení – efektivně se jedná o jedno velké pole. Stack je další úložiště, kde ale můžeš manipulovat pouze s prvky na vrchu – tedy poslední přidaná data.²

Stroj pracuje s neomezeně velkými³ integery, tedy celými čísly se znaménkem. Na začátku dostaneš data na stacku a v paměti, a očekává se, že tvůj algoritmus zvládne upravit tyto data tak, aby odpovídaly očekávanému výstupu.

Interpreter přijímá text jako zdrojový kód. Je ve formátu jedné instrukce na jeden řádek, prázdné řádky se ignorují. Komentáře lze psát pomocí znaku # a je možné pro skoky používat labely. Tedy je možné zadefinovat label tím, že se někam napíše `jmenoLabelu:` a ve skokových instrukcích lze nahradit jakýkoliv argument labellem. Ten se před vykonáním instrukce převede na odpovídající číslo. Lze tedy napsat buď `JMP 5`, nebo třeba `JMP mojeSuperFunkce`.

¹<https://github.com/fiks-org/fikslang-interpreter/blob/main/INSTRUCTIONS.md>

²Samozřejmě se vám podařilo vymyslet instrukce, které to do nějaké míry obcházejí.

³V rámci mezí.

Příklad programu

Následující příklad vezme číslo, a pokud je liché, přičte k němu jedničku.

```
PUSH 11
PROC roundToEven
EXIT

incSecondValue:
    ROTATE 3
    INC
    ROTATE 3
RET

roundToEven:
    SWAP
    DUP
    PUSH 2
    SWAP
    MODULO
    SKIP_IF EQ0
        PROC incSecondValue
    SWAP
    POP2
    SWAP
RET
```

Tento příklad obsahuje použití funkcí. Doporučím se podívat do seznamu instrukcí, kde je popsáno jak funkce fungují, případně nahlédnout i do samotné implementace. Při řešení této úlohy se může hodit vědět, jak s nimi pracovat.

Zadání

Tvým úkolem bude vytvořit program, který bude automatizovat část práce, kterou posádka musela doteď dělat pracně manuálně: porovnávat textové řetězce. Na stacku dostane na začátku dva prvky: indexy do paměti, kde začínají textové řetězce. V paměti jsou tyto řetězce reprezentované jako čísla, konkrétně ASCII hodnoty daných znaků. Každý z řetězců je ukončen nulou.

Tvým cílem bude spočítat počet indexů, na kterých se tyto dva řetězce liší. Tento výsledek je na konci nutné mít na stacku na nejvrchnější pozici, všechnen ostatní obsah stacku a paměti se bude ignorovat.

Tedy například, pro FIKS a FIT bude výsledek 2. Na třetí pozici se liší písmena *K* a *T*, a na čtvrté pozici přebývá písmeno *S*. Obodobně, pro text ABC a prázdný řetězec bude výsledek 3.

Hodnocení

- 2 body, pokud správně vyřešíš příklady, kde jsou řetězce stejně dlouhé.
- 4 body, pokud navíc správně vyřešíš příklady, kde jsou řetězce různě dlouhé, nebo i prázdné.
- 5 bodů, pokud vyřešíš i obzvlášť zákeřné příklady.
- 8 bodů, pokud v programu nebudeš používat instrukci INC, ale buď se jí úplně vyhneš, nebo si ji implementuješ sám.

Odevzdávání

Úloha se odevzdává jako sfinga. Na webu si nejdříve vygeneruješ zadání, které obsahuje ID tvého problému, ty ho ovšem nijak nemusíš brát v potaz. Jediné, co odevzdáváš, je tvůj zdrojový kód. Po odevzdání se spustí na serveru a ty dostaneš své body.

Příklad

Vstup:

Stack	Paměť	
	Index	Hodnota
1 6	1	70
	2	73
	3	75
	4	83
	5	0
	6	70
	7	73
	8	84
	9	0

Výstup:

Stack
2
...