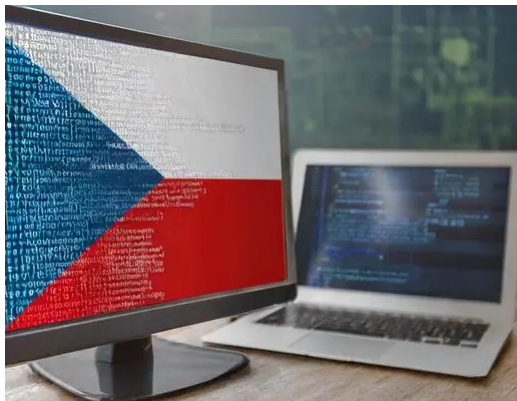


Mateřský programovací jazyk



Tato úloha obsahuje vypisování desetinných čísel. Pro správné vyhodnocení systémem je nutné tato čísla vypisovat bez jakýchkoli přebytečných koncových nul. Jak to zajistit naleznete v šestém bodě [technických pravidel](#).

Je potřeba spustit program napsaný v jazyce Č, jehož klíčová slova jsou přejata z češtiny.

V Č lze užitím klíčového slova `je` zakládat číselné proměnné nebo existující proměnným. Jejich název může obsahovat písmena anglické abecedy a číslice, začínat ale musí písmenem. Datový typ se neuvádí. Čísla mohou být celá nebo desetinná, oddělovačem je pak tečka. Příklad:

```
pocetKnedliku je 5
objemOmacky je 1.74
```

K dispozici jsou operátory pro sčítání `a` a pro odečítání `bez`. Příklad:

```
pocetKnedlikuSnedenych je 4
pocetKnedliku je 3 a 2 bez pocetKnedlikuSnedenych
```

`je` může také změnit hodnotu již existující proměnné. Příklad:

```
objemOmacky je 5.33
objemOmacky je 5
objemOmacky je 4.8
```

Jazyk podporuje podmínky pomocí `kdyz` a `tak` s logickými operátory `rovno`, `aspon` (větší než nebo rovno) a `nejvic` (menší než nebo rovno). Logické výrazy jde řetězit operátorem `nebo`, takže pak pro splnění podmínky stačí, aby platil jeden z nich. Podmíněné řádky jsou odsazeny dvojicí mezer. Podmínky je možné zanořovat. Příklad:

```
pocetKnedliku je 5
objemOmacky je 1.74
urovenHladu je 2

kdyz pocetKnedliku aspon 2 tak
    urovenHladu je 1
    kdyz pocetKnedliku aspon 4 nebo objemOmacky aspon 0.3 tak
        urovenHladu je 0
```

Č disponuje také cyklem `dokud - tak`, který se opakuje, dokud uvedená podmínka platí. I zde lze využít `nebo`. Cykly mohou být umístěny v podmínkách (a naopak), ale nikdy jich není vnořeno více do sebe. Příklad:

```
pocetKnedliku je 5
objemOmacky je 1.74

dokud pocetKnedliku aspon 1 tak
    pocetKnedliku je pocetKnedliku bez 1

    kdyz objemOmacky aspon 0.1 tak
        objemOmacky je objemOmacky bez 0.1
```

Narozdíl od jiných jazyků, v Č jsou proměnné vytvořené v podmínce nebo cyklu dostupné i později. Po skončení dané podmínky či cyklu takové proměnné nezanikají a lze je číst. Pokud ale podmínka nebyla splněna, a proměnná v ní tak ani nevznikla, a přesto dojde k pokusu o její přečtení, nastává chyba.

Jazyk umožňuje vypisovat hodnoty pomocí `vypsati`. Příklad:

```
pocetKnedliku je 5
objemOmacky je 1.74

vypsati 3
vypsati pocetKnedliku
vypsati objemOmacky a 0.3
```

A konečně, kód napsaný v Č je možné komentovat užitím `//`. Jakýkoli text na řádce za touto sekvencí je při vykonávání programu přehlížen. Příklad:

```
// tohle se neprovede
// pocetKnedliku je 5
objemOmacky je 1.74 // mnam
```

Na vstupu programu bude (na samostatných řádcích):

- počet řádků kódu `c` (celé číslo od 0 do 1 000),
- `c` řádků kódu v jazyce Č.

Zadaný program nebude nikdy nekonečný. Kód bude obsahovat pouze znaky anglické abecedy, čísla a mezery. Nebudou v něm chyby v odsazení nebo přebytné mezery.

Úkolem je tento kód takzvaně interpretovat, tedy řádek po řádku vykonat. Cokoli by program vypsal skrze `vypsat`, má vypsat také vaše řešení na svůj výstup.

Při běhu programu mohou nastat chyby:

- použití neexistující proměnné,
- pojmenování proměnné některým z klíčových slov,
- nesprávné užití klíčového slova (např. `je` bez předané hodnoty nebo `kdyz`, kterému chybí `tak`).

Pokud by v programu nastala některá z uvedených chyb, interpretace bude ukončena a nesmí být vypsáno nic (ani to, co by se vypsal ještě před chybou). Pouze se vypíše hláška `error`.

Kdyby chyba nastala v podmínce nebo cyklu, do kterého se program při interpretaci ale nedostal, nevádí to a vykonávání pokračuje dále.

Příklad #1

Vstup

```
18
x je 5
y je -1
z je x a y bez 3

kdyz y aspon 0 nebo x rovno 4 nebo z nejvic 1 tak
  p je 4

dokud x bez 1 aspon y a z tak
  x je x bez 1
  kdyz y aspon -3 tak
    y je y bez 1
    p je p a 1
    vypsat x

kdyz z rovno 10 tak
  vypsat neexistujciPromenna // sem se program nedostane takze nevadi

vypsat p a z bez 2
```

Výstup

```
4
3
2
6
```

Příklad #2

Vstup

```
16
cislo je 5
vypsati cislo

vypsati cislo2 // promenna neexistuje

kdyz cislo aspon 10
    cislo3 je 2
vypsati cislo3 // promenna neexistuje

nebo je 5 // chybne pojmenovani klicovym slovem

kdyz cislo rovno 6 // chybi slovo tak
    vypsati 3

kdyz cislo nebo cislo2 rovno aspon 6 nebo tak // nesmyslny logicky vyraz
    vypsati 3
```

Výstup

```
error
```

Veškeré výukové materiály se nachází na <https://github.com/delta-cs/seminar>.

V případě, že vám řešení nebude uznáno a nebudete vědět proč, navštivte nejdříve [stránku s technickými pravidly](#) a ujistěte se, že program všechna splňuje.