

Vyhodnocování řetězcových výrazů

Je dáno několik operací nad řetězci a z nich je sestaven krátký lineární program podle pravidel daných níže. Zápis programu je na vstupu a úlohou je zjistit, jakou řetězcovou hodnotu vrací funkce na posledním řádku programu.

Primitiva

`<id_proměnné>` je malé písmeno z množiny `{'a', ..., 'z'}`, proměnných tedy může být nejvýše 26. Proměnné jsou výhradně řetězcové, jiné hodnoty do nich ukládat nelze.

`<řetězcová konstanta>` je posloupnost libovolných tisknutelných znaků ASCII (32 až 127) kromě znaku úvozovek uzavřená v úvozovkách (př.: `"C4"`, `"yz_ab"`), prázdný řetězec má standardní zápis `""`. Lomítková konvence jako např. `\n` `\u0123` apod. se nepoužívá. Znak v řetězci se indexují od 0.

`<číselná konstanta>` je neprázdná posloupnost číslic z množiny `{'0', '1', '2', ..., '9'}`.

Funkce

`<řetězcová funkce>` implementuje jednu ze tří operací Insert, Delete, Extract, které vrací řetězec.

`<číselná funkce>` implementuje jednu ze dvou operací Position, Length, které vrací celé číslo.

Funkce Insert

`<řetězcová funkce> = I(<kam_str>, <kam_index>, <co_str>)`

`<kam_str>`, `<co_str>` je `<řetězcová konstanta>` nebo `<řetězcová funkce>` nebo `<id_proměnné>`

`<kam_index>` je `<číselná konstanta>` nebo `<číselná funkce>`

Funkce I vrací řetězec, který vznikne vložením řetězce `<co_str>` do řetězce `<kam_str>` mezi znaky na pozicích `<kam_index>-1` a `<kam_index>`,

Pokud je hodnota `<kam_index>` menší nebo rovna nule, vkládá funkce I řetězec `<co_str>` na začátek řetězce `<kam_str>`. Pokud je hodnota `<kam_index>` větší nebo rovna délce řetězce `<kam_str>`, vkládá funkce I řetězec `<co_str>` na konec řetězce `<kam_str>`. Příklad: `I("abc", 2, I("13", 1, "222"))` vrací `ab12223c`.

Funkce Delete

`<řetězcová funkce> = D(<kde_str>, <od_index>, <do_index>)`

`<odkud_str>` je `<řetězcová konstanta>` nebo `<řetězcová funkce>` nebo `<id_proměnné>`

`<od_index>`, `<do_index>` je `<číselná konstanta>` nebo `<číselná funkce>`

Funkce D vrací řetězec, který vznikne z řetězce `<kde_str>` smazáním podřetězce začínajícího na pozici s indexem `<od_index>` a končícím na pozici s indexem `<do_index>`.

Pokud je hodnota `<od_index>` větší než hodnota `<do_index>` nebo pokud je řetězec `<kde_str>` prázdný, funkce D vrátí řetězec `<kde_str>` beze změny. Pokud jsou obě hodnoty `<od_index>`, `<do_index>` současně větší nebo rovny délce řetězce `<kde_str>` nebo pokud jsou současně obě menší než nula, vrací funkce D řetězec `<kde_str>` beze změny. V ostatních případech, pokud je hodnota `<od_index>` menší než nula, použije funkce D místo ní hodnotu 0, pokud je hodnota `<do_index>` větší nebo rovna délce řetězce `<kde_str>`, použije funkce D místo ní index posledního znaku v řetězci `<kde_str>`.

Příklad: `D(D("abcdefghij", 2, 3), 3, 4)` vrací `abehij`.

Funkce Extract

`<řetězcová funkce> = E(<odkud_str>, <od_index>, <do_index>)`

`<odkud_str>` je buď `<řetězcová konstanta>` nebo `<řetězcová funkce>` nebo `<id_proměnné>`

`<od_index>`, `<do_index>` je `<číselná konstanta>` nebo `<číselná funkce>`

Funkce E vrací podřetězec řetězce `<odkud_str>`, jehož první znak leží na pozici s indexem `<od_index>` a poslední znak na pozici s indexem `<do_index>`.

Pokud je hodnota `<od_index>` větší než hodnota `<do_index>` nebo pokud je řetězec `<odkud_str>` prázdný, vrací funkce E prázdný řetězec. Pokud jsou obě hodnoty `<od_index>`, `<do_index>` současně větší nebo rovny délce řetězce `<odkud_str>` nebo pokud jsou současně obě menší než nula, vrací funkce E prázdný řetězec. V

ostatních případech, pokud je hodnota `<od_index>` menší než nula, použije funkce `E` místo ní hodnotu 0, pokud je hodnota `<do_index>` větší nebo rovna délce řetězce `<odkud_str>`, použije funkce `E` místo ní index posledního znaku v řetězci `<odkud_str>`. Příklad: `E(E("abcdefghij",1,7),0,2)` vrací `bcd`.

Funkce Position

`<číselná funkce> = P(<včem_str>,<čeho_str>)`

`<včem_str>`, `<čeho_str>` je `<řetězcová konstanta>` nebo `<řetězcová funkce>` nebo `<id_proměnné>`
Funkce `P` vrací celé číslo, které je rovno indexu pozice prvního výskytu řetězce `<čeho_str>` v řetězci `<včem_str>`. Pokud se řetězec `<čeho_str>` v řetězci `<včem_str>` nevyskytuje, vrací funkce `P` hodnotu `-1`. Pozice prázdného řetězce v jakémkoli řetězci je 0. Příklad: `P("bbabacaba","ab")` vrací 2.

Funkce Length

`<číselná funkce> = L(<čeho_str>)`

`<čeho_str>` je `<řetězcová konstanta>` nebo `<řetězcová funkce>` nebo `<id_proměnné>`
Funkce `L` vrací celé nezáporné číslo, které je délkou řetězce `<čeho_str>`. Délka prázdného řetězce je standardně 0. Příklad: `L("aaabbb")` vrací 6.

Program

Program se skládá z přiřazovacích příkazů, přičemž na každém řádku je právě jeden příkaz.

Všechny příkazy kromě posledního mají tvar

`<identifikátor proměnné>=<řetězcová konstanta>`, (např. `x="uw123"`)

poslední příkaz má tvar

`<identifikátor proměnné>=<řetězcová funkce>`

Program obsahuje vždy alespoň jeden příkaz, pokud je příkaz právě jeden, považujeme jej za poslední. Identifikátory proměnných se v programu na levé straně neopakují, tím je délka programu omezena na nejvýše 26 příkazů. Středníky ani žádné jiné oddělovače příkazů (kromě implicitního konce řádku) se v programu nepoužívají. Mezery mají význam jen jako součást řetězcových konstant, jinde v programu se nevyskytují. Program je napsán syntakticky správně, jeho syntaxi není třeba kontrolovat.

Vstup

Program ve formátu popsaném výše, vstupní soubor neobsahuje prázdné řádky.

Výstup

Hodnota proměnné na posledním řádku nalevo v přiřazovacím příkazu po provedení všech příkazů v programu. Tato hodnota je na výstupu uzavřena do úvozovek

Příklad

Vstup:

```
x="23"
```

```
c="yzab"
```

```
w=I(I(I(D(c,2,L(c)),0,x),0,E(c,2,L(c))),L(x),"01")
```

Výstup:

```
"ab0123yz"
```

Vstup:

```
e="udub"
```

```
f="ububu"
```

```
g="o"
```

```
h="obudo"
```

```
k=I(D(I("udodu",P("odod","udob"),g),L("odud"),4),4,E(E("ubu",3,5),L("udododudu"),8))
```

Výstup:

```
"oudou"
```