

ZADÁNÍ SEMESTRÁLNÍ PRÁCE

INTERPRET PODMNOŽINY JAZYKA LISP

Zadání

Naprogramujte v ANSI C přenositelnou¹ **konzolovou aplikaci**, která bude fungovat jako jednoduchý interpret podmnožiny jazyka **LISP**. Vstupem aplikace budou příkazy v jazyce **LISP**. Výstupem pak výsledek vyhodnocení každého výrazu.

Program se bude spouštět příkazem `lisp.exe [<vstupní-soubor>]`. Symbol `<vstupní-soubor>` zastupuje nepovinný parametr – název vstupního souboru se seznamem výrazů v jazyce LISP. Není-li první parametr uveden, program bude fungovat v interaktivním módu, kdy se příkazy budou provádět přímo zadáním z konzole do programu.

Úkolem Vámi vyvinutého programu tedy je:

1. Při spuštění bez parametru bude čekat na vstup od uživatele. Zadaný výraz vyhodnotí a bude vyžadovat další vstup, dokud nebude uveden výraz (`quit`).
2. Při spuštění s parametrem načte zadaný vstupní soubor, každý výraz v něm uvedený vypíše na obrazovku, okamžitě vyhodnotí a výsledek vypíše na obrazovku. Po zpracování posledního výrazu dojde k ukončení programu. Proto nemusí být jako poslední výraz uveden výraz (`quit`). Na jedné řádce v souboru může být uvedeno více samostatných výrazů a program je musí být schopen správně zpracovat.

Váš program může být během testování spuštěn například takto:

```
lisp.exe test.lisp
```

Hotovou práci odevzdejte uploadem na server podle pokynů na webu, v jediném archivu ZIP, pojmenovaném Vaším osobním číslem, tj. např. `A09B1234P.zip`. Archiv nechtě obsahuje všechny zdrojové soubory potřebné k přeložení programu, `makefile` pro Windows i Linux (pro překlad v UNIXu/Linuxu připravte soubor pojmenovaný `makefile` a pro Windows `makefile.win`) a dokumentaci ve formátu PDF vytvořenou v typografickém systému $\text{T}_{\text{E}}\text{X}$, resp. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Bude-li některá z částí chybět, kontrolní skript Vaši práci odmítne.

Popis činnosti programu

Program zpracuje postupně zadané výrazy v jazyce LISP, po zadání výraz vyhodnotí a jeho výsledek okamžitě vypíše na obrazovku.

Specifikace vyhodnocovaných výrazů

Výrazem mohou být:

konstanta Celočíslná hodnota. Její vyhodnocení vrací přímo danou hodnotu (vyhodnocení samo na sebe).

¹Je třeba, aby bylo možné Váš program přeložit a spustit na PC s operačním prostředím Win32/64 (tj. operační systémy Microsoft Windows NT/2000/XP/Vista/7) a s běžnými distribucemi Linuxu (např. Ubuntu, OpenSUSE, Gentoo, Debian, atd.). Server, na který budete Vaši práci nahrávat a který ji otestuje, má nainstalovaný operační systém Debian GNU/Linux 5.0 s překladačem gcc 4.4.4.

proměnná Název, pod kterým je uložena nějaká celočíselná hodnota. Její vyhodnocení vrací uloženou hodnotu (ukládání se bude provádět pomocí funkce `set`).

seznam Seznam výrazů je dán obsahem kulatých závorek. Výrazy lze libovolně vnořovat. Prvním prvem v seznamu je vždy operátor nebo název funkce. Program bude podporovat minimálně operátory a funkce dané tabulkou ¹.

Veškeré názvy a textové konstanty jsou ukládány vždy s **velkými písmeny**.

Specifikace výstupu programu

Program bude vždy vypisovat výsledek právě zadaného výrazu. Při načtení seznamu příkazů ze souboru musí výstup odpovídat přesně výstupu spuštění programu bez parametru (interaktivní režim).

Chybová hlášení

Program vypíše chybové hlášení, je-li ve vyhodnocovaném výrazu alespoň jedna z následujících chyb:

- První prvek seznamu není operátor nebo funkce. Vyjímku tvoří funkce `quote` a `list`, které nenechají seznam vyhodnotit.
- Jeden nebo více ostatních prvků (tedy kromě prvního) seznamu není konstanta, seznam, existující proměnná a nebo není uvedena jako `quote`.
- Použití prázdného seznamu (tj. hodnoty `NIL`) jako operandu. Hodnota `NIL` není číslo.

Užitečné techniky a odkazy

Uvedené techniky je možné (ale nikoliv nezbytně nutné) využít při řešení úlohy. Protože se jedná o postupy víceméně standardní, lze k nim nalézt velké množství dokumentace:

1. **S-expression**
<http://en.wikipedia.org/wiki/S-expression>
2. **Rekurzivní sestup**
3. **Interpret LISPu** – pro porovnání správného chování Vámi vytvořeného interpretu.

Řešení úlohy je zcela ve vaší kompetenci – zvolte takové algoritmy a techniky, které podle vás nejlépe povedou k cíli . . .

Příloha 1 – Ukázka interpretace

```
[1] > (+ 4 6 (* 2 6 7 (+ 5 2)))
598
[2] > ()
NIL
[3] > (set 'a 10)
10
[4] > (+ 3 a)
13
```

²Podpora dalších operátorů či funkcí je jen na studentovi.

```

[5]> (= 1 (- 2 1))
T
[6]> (/= 2 (/ 4 2))
NIL
[7]> '(a b (c d))
(A B (C D))
[8]> (car (list 1 2 3))
1
[9]> (cdr (list 2 3 5))
(3 5)
[10]> (car (list (list 1 2)))
(1 2)
[11]> 5
5
[12]> a
10
[13]> (quit)
Bye.

```

Příloha 2 – Příklad zdrojového souboru

Uvedený obsah souboru odpovídá seznamu příkazů v příloze 1 (jen chybí příkaz `(quit)`).

```

(+ 4 6 (* 2 6 7 (+ 5 2)))
()
(set 'a 10) (+ 3 a)
(= 1 (- 2 1))
(/= 2 (/ 4 2))
'(a b (c d))
(car (list 1 2 3))
(cdr (list 2 3 5))
(car (list (list 1 2)))
5 a

```

Tabulka 1: Podporované operátory a funkce

Operátor / Funkce	Význam	Příklad	Výsledek
+	Sečte prvky a vrátí celočíselný výsledek	(+ 1 2 5)	8
-	Odečte prvky zleva doprava a vrátí celočíselný výsledek	(- 5 3)	2
*	Vynásobí prvky a vrátí celočíselný výsledek	(* 6 2)	12
/	Vydělí od sebe prvky zleva doprava a vrátí celočíselný výsledek. Prováděno bude vždy jen celočíselné dělení.	(/ 8 2) (/ 3 2)	4 1
=	Porovná sousední prvky na shodu a výsledkem je T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(= 2 2) (= 3 1)	T NIL
/=	Porovná sousední prvky na neshodu a výsledkem je T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(/= 3 3) (/= 3 2 2) (/= 3 2 1)	NIL NIL T
<	Porovná sousední prvky operátorem < (menší než) a vrátí výsledek T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(< 3 1 4) (< 1 2 3)	NIL T
>	Porovná sousední prvky operátorem > (větší než) a vrátí výsledek T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(> 3 2 1) (> 4 1 5)	T NIL
<=	Porovná sousední prvky operátorem > (menší nebo rovno než) a vrátí výsledek T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(<= 2 2 2 4) (<= 3 2)	T NIL
>=	Porovná sousední prvky operátorem > (větší nebo rovno než) a vrátí výsledek T nebo NIL. Výsledek je dán logickým součinem všech porovnáání.	(>= 2 2 2 4) (>= 3 2)	NIL T
'	Stejné chování jako funkce quote	'(a b 2)	(A B 2)
quote	Argument nebude vyhodnocen	(quote A) (quote (a b))	A (A B)
list	Argumenty funkce budou vráceny jako seznam	(list 1 2 3) (list 1 (list 2 3))	(1 2 3) (1 (2 3))
quit	Ukončí program	(quit)	-
car	Vrátí první prvek zadaného seznamu	(1 2 3 4)	1
cdr	Vrátí zadaný seznam bez prvního prvku	(1 2 3 4)	(2 3 4)
set	Uloží do proměnné dané prvním argumentem hodnotu danou druhým argumentem. Funkce vrací ukládanou hodnotu. Uložená hodnota může být použita v dalších výrazech.	(set 'a 4)	4