

Prevođenje programskih jezika - Januar 1, 27.01.2019.

Napomena: Zadatke čuvajte u direktorijumu formata

PPJ.jan1.2019.ime.prezime.indeks.godina u zasebnim direktorijumima 1 i 2.

Na primer, student Marko Markovic sa indeksom 27/2013, prvi zadatak čuva u poddirektorijumu 1 direktorijuma PPJ.jan1.2019.Marko.Markovic.27.2013.

Samo sintaksno ispravan kod može nositi više od 0 poena. Makefile je obavezan deo rešenja.

1. Napisati interpretator za minijturni programski jezik koji podržava rad sa označenim 32-bitnim celim brojevima. Brojevne konstante se mogu navoditi zapisane u dekadnom, oktalnom i heksadekadnom brojevnom sistemu. Dekadne konstante se zapisuju uobičajeno, heksadekadne sa prefiksom 0x, a oktalne sa prefiksom 0. Jezik raspolaže svim aritmetičkim (+, -, *, /) i bitskim operatorima (&, |, ^, ~, <<, >>) i relacionim operatorom ==. Svi su po uzoru na operatore iz jezika C.

Funkcija *print(x, b)* ispisuje brojne vrednosti u osnovi *b*. Pri čemu osnova može biti samo neka od podržanih zapisa za konstante.

Moguće je korišćenje promenljivih, ali svaka mora biti deklarirana pre korišćenja. Imena promenljivih počinju znakom `_` iza koje obavezno sledi slovo, iza koga mogu i ne moraju se javiti slova ili cifre.

Programski jezik podržava i rad sa listama označenih celih brojeva. Podržane su operacije ispisa elementa liste i pristup elementima preko indeksa. Identifikator liste počinje uvek malim slovom `l` koje je praćeno sa bar jednim slovom, cifrom ili `_`.

Svaka greška uzrokuje ispis poruke na ekran i prekid programa. Svaka naredba programskog jezika se završava sa `;`.

Test primeri sa izlazom u produžetku:

```
(a)    int _a = 0xFF;
        print(_a, 8);           977
        print(_a);              255
        print(_a, 16);          FF
        print(_a, 19);          Nepodrzana osnova!
        print(_nova1);          _nova1 nije definisana!

(b)    int _a = 2;
        int _b = _a + 1;
        _b = (_a + 1) << (_a * 2);
        print(_b + 2, 16);      32
        int _c ;
        _c = (_a + _b) / 012 + 013;
        print(_c, 16);          10
        _a = 7;
        _b = ~(_a - 1);
        _b & _a == 0;           False!
        _a == _a | 0x0 & 0xFFFFFFFF; True!
        int _nova2 = _c >> (_a - 1 ~ _b); Drugi operand mora biti veci od nule!

(c)    list lx = [-1, 02, 0x3];
        print( lx);              [ -1, 2, 3 ]
        print(lx[0] << lx[1]);   -4
```

2. Koristeći tehniku rekurzivnog spusta implementirati sintaksni analizator koji prepoznaje izraze nad označenim brojevima korišćenjem samo bitskih operatora (&, |, ~) i dodelu promenljivoj, kao i ispis promenljive koristeći funkciju `print`, bez navodjenja osnove. Nije potrebno vršiti izračunavanja tokom parsiranja ulaza već samo proveriti sintaksu.

Pomoć:

- `std::vector`

- `size_type size() const;`
- `iterator begin(); iterator end();`
- `iterator insert(iterator pos, const T& value);`
- `iterator erase(iterator pos);`
- `void push_back(const T& value);`
- `void pop_back();`
- `reference operator[] (size_type pos);`
`const_reference operator[] (size_type pos) const;`

- `std::map`

- `std::pair<iterator,bool> insert(const value_type& value);`
Member type `value_type` is the type of the elements in the container, defined in `map` as `pair<const key_type,mapped_type>`
- `void erase(iterator pos);`
- `iterator find(const Key& key);`