Prevođenje programskih jezika

Na Desktopu napraviti direktorijum sa nazivom **PPJ.jun1.ime.prezime.brojIndeksa.godina** i u njemu sačuvati svoj rad. Dakle, ako **Pera Perić** sa indeksom **123/2015** polaže ispit potrebno je da na Desktopu napravi direktorijum sa nazivom **PPJ.jun1.Pera.Peric.123.2015** i u njemu da sačuva svoj rad. **Sintaksno neispravni zadaci se ne pregledaju. Vreme za izradu ispita je 3h. Uslov za polaganje ispita je minimum 20p.**

Sintaksna analiza naviše

Napredne strukture podataka poput steka i reda se na kursu Programiranju 2 samo spominju. Pri kraju semestra nam je ostalo dovoljno vremena da im te koncepte malo bolje predstavimo. Potreban nam je, u te svrhe, interpreter koji će nam omogućiti sledeće:

1. Interpreter treba da omogući jednostavno definisanje stekova i redova, prikazivanje njihovog sadržaja i operaciju dodele. Stekovi i redovi se mogu definisati samo nad celim brojevima i potrebno ih je definisati kao u navedenim primerima. Stekovi i redovi mogu biti proizvoljne dužine. Funkcija *print_stack* Štampa elemente steka od elementa na dnu do elementa na vrhu steka, a funkcija *print_queue* Štampa elemente reda od prvog elementa u redu do elementa na kraju reda. U slučaju da se greška dogodi, potrebno je obavestiti korisnika i prekinuti sa izvršavanjem. Imena promenljivih moraju počinjati znakom _.

```
stack<int> _x := {1, 2, 3};
queue<int> _y := {2, 4, 6};
stack<int> _z;                          // Definiše prazan stek!
_z := _x;
_z := _y;                               // Nije moguće izvršiti dodelu!
print_stack(_x);                        [1, 2, 3]
print_queue(_y);                        [2, 4, 6]
print_queue(_x);                        //Greska! _x je deklarisano kao stek!
queue<number> _x := {5, 5, 5};          // Greska! _x je već definisano!
```

2. Interpreter treba da omogući i neke operacije nad stekovima i redovima. Metod *push*(x) dodaje element *x* na kraj reda, odnosno na vrh steka. Ekvivalentno, funkcija *pop()* uklanja element sa vrha steka, odnosno početka reda. Metoda *peek*() vraća element sa vrha steka (odnosno početka reda). Metod *size()* vraća broj elemenata u steku/redu.

```
stack<int> _w;
_w.push(5);
_w.push(9);
print_stack(_w);                        [5, 9]
_w.peek();                              [9]
_w.pop(); print_stack(_w);              [5]
print_queue(y);                         [2, 4, 6]
_y.pop(); print_queue(_y)               [4, 6]
x.size();                               1
y.size();                               2
y.peek();                               4
```

3. Interpreter treba da omogući osnovne aritmetičke operacije sa sledećim značenjem: + - nadovezivanje sadržaja drugog sabirka na sadržaj prvog sabirka (sadržaj nadovezivati onim redosledom kojim bi bili ispisani), unarni minus vrši obrtanje sadržaja steka/reda, () – grupisanje.

```
stack<int> _k := {1, 2, 3};
stack<int> _l := {6, 7, 2, 3};
```

```
queue<int> _s := {1, 5};
queue<int> _t := {6, 4};
queue<int> _test:= _s + _t; print_queue(_test);          [1, 5, 6, 4]
test:=t+(-s); print_queue(test);                         [6, 4, 5, 1]
_k:=-k; print_stack(_k);                                 [3, 2, 1]
_l:=_l+_l; print_stack(_l);                              [6, 7, 2, 3, 6, 7, 2, 3]
```

4. Interpreter treba da omogući upoređivanje odgovarajućih struktura podataka (==, !=). Stekovi se upoređuju od vrha ka dnu, a redovi od početka ka kraju.

```
_k:={2, 4, 6};
_l:={2, 4, 6, 8};
_k==_l;                                                  False!
_k!=_l;                                                  True!
_l.pop();
_k==_l;                                                  True!

_s:={1,3,5};
_t:={3,5};
_s!=_t;                                                  True!
_s.pop();
print_queue(_s);                                         [3, 5]
_s==_t;                                                  True!
```

Napomena: Makefile je obavazen deo rešenja.

Sintaksna analiza naniže

Uz pomoć rekurzivnog spusta implementirati sintaksni analizator koji omogućava proveru sintaksne ispravnosti dela pod 1 iz prethodnog zadatka (ali samo za stekove). U komentaru programa je neophodno ostaviti sređenu gramatiku i skupove izbora.

```
stack<int> _x;
stack<int> _y := {1,2,3,4,5,6};
stack<int> _z := _x;
_y := _z;
print_stack(_x);
print_stack(_w);
```

Ispit traje 3h.