

## Úloha: převod hradlové sítě do jednoduchého assembleru

Je daná hradlová síť jako DAG (Directed Acyclic Graph), kde vrcholy jsou hradla s jedním (not) nebo dvěma vstupy (and, or) a listy (list zde chápeme jako vrchol, do něž nevede žádná orientovaná hrana) jsou názvy vstupních registrů typu bool. Celá síť má pouze jeden kořen (kořen je zde vrchol, ze kterého nevede žádná orientovaná hrana).

Dále je dán jednoduchý assembler s následujícími instrukcemi, kde v závorkách je uvedena sémantika každé instrukce:

NOT Reg1, Reg2; (Reg1 := not Reg2)

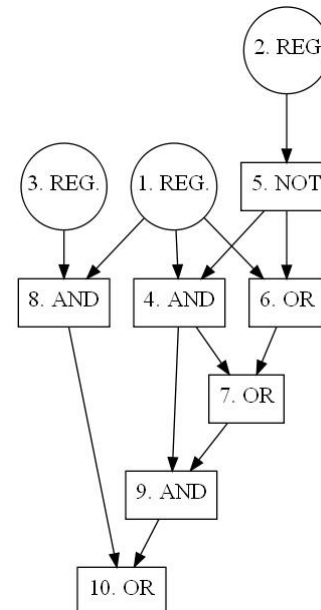
AND Reg1, Reg2, Reg3; (Reg1 := Reg2 and Reg3)

OR Reg1, Reg2, Reg3; (Reg1 := Reg2 or Reg3)

A boolovské registry R1, R2, ... a pomocné boolovské registry T1, T2 ... (pomocné registry se v zadané hradlové síti nevyskytují).

Úkolem je vygenerovat assemblerovský program, který spočítá zadanou hradlovou síť a výsledek uloží do registru T1.

Příklad:



Vstup:

```
10 - počet vrcholů
R - registr
R
R
A - hradlo AND
N - hradlo NOT
O - hradlo OR
O
A
A
O
13 - počet hran
1 4 - hrana z 1. Vrcholu do 4.
5 4
2 5
1 6
5 6
6 7
4 7
1 8
3 8
4 9
7 9
9 10
8 10
```

Výstup:

```
NOT T2, R2;
AND T3, R3, R1;
AND T4, R1, T2;
OR T5, R1, T2;
OR T2, T4, T5;
AND T5, T4, T2;
OR T1, T3, T5;
```