

# 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 (AND, OR) vstupy a kořeny (kořen chápeme jako vrchol, do něžž nevede žádná orientovaná hrana) jsou názvy vstupních registrů typu bool ( $R_1, R_2, \dots, R_N$ ). Celá síť má pouze jeden list (list je vrchol, ze kterého nevede žádná orientovaná hrana). Výstupem hradlové sítě pro libovolné hodnoty vstupních registrů je hodnota hradla v listu sítě.

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

```
NOT Reg1,Reg2;      (Reg1 := not Reg2)
AND Reg1,Reg2,Reg3; (Reg1 := Reg2 and Reg3)
OR  Reg1,Reg2,Reg3; (Reg1 := Reg2 or Reg3)
```

## Úkol

Úkolem je vygenerovat assemblerovský program, který simuluje zadanou hradlovou síť, t.j. na základě hodnot vstupních registrů vypočte hodnotu hradla v listu sítě. Assembler má k dispozici neomezený počet pomocných registrů typu bool ( $T_1, T_2, \dots$ ). Program v assembleru uloží výsledek do registru T1.

## Formát vstupu

Hradlová síť je popsána ve vstupním souboru, který program dostane na standardním vstupu. Vstupní soubor se skládá z dvou částí seznamu vrcholů a seznamu hran. První řádek seznamu je počet prvků. Vrcholy jsou čtyř typů: R - vstupní registr, N - hradlo NOT, O - hradlo OR, A - hradlo AND. Vrcholy jsou číslovány od 1 podle pořadí v seznamu vrcholů, každý řádek obsahuje jeden vrchol. Hrany jsou dány jako dvojice čísel vrcholů oddělených mezerou, jedna hrana na řádku.

## Formát výstupu

Výstupem je program v assembleru, jedna instrukce assembleru na řádku. Řešení zapíšte na standardní výstup. Hradlovou síť lze často přeložit do assembleru více různými způsoby (mohou se např. lišit pořadím některých příkazů), vyhodnocovací systém ale každé výsledkově ekvivaletní řešení vyhodnotí jako korektní.

## Pozor na pořadí vrcholů

Ukazuje se, že v ukázkových i v testovacích souborech byly původně vrcholy vepsány již v topologickém uspořádání a stačilo je tedy jen načíst a pak ve stejném nebo obráceném pořadí vypsát v podobě assemblerovských příkazů. Někteří řešitelé si toho všimli, využili a dodali kód, který úlohu řeší skutečně jen pro tento velmi speciální případ, kdy jsou vstupní data už předem uspořádána v potřebném pořadí.

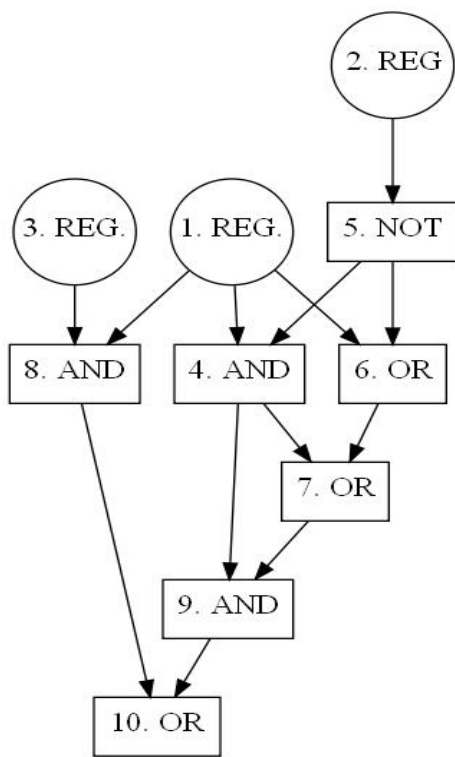
Data jsme tedy nyní modifikovali, to jest zachovali jsme strukturu testovacích sítí, jen jsme náhodně změnili pořadí vrcholů ve vstupních datech (viz ukázka číslo dvě).

### Příklad hradlové sítě

#### Vstup:

#### Výstup:

```
NOT T4,R2;
AND T3,R1,T4;
OR  T6,R1,T4;
OR  T5,T6,T3;
AND T2,T3,T5;
AND T7,R1,R3;
OR  T1,T2,T7;
```



10 - počet vrcholů

R - registr

R

R

A - hradlo AND

N - hradlo NOT

O - hradlo OR

O

A

A

O

**Další příklad, vstup:**

1 4 - hrana z 1. Vrcholu do 4.

15

R

O

R

O

R

R

A

R

R

A

O

O

R

O

14

13 7

4 7

2 13

15 13

11 4

12 4

1 2

14 2

5 15

9 15

6 11

8 11

3 12

10 12

## **Výstup**

OR T3,R1,R14;

OR T4,R5,R9;

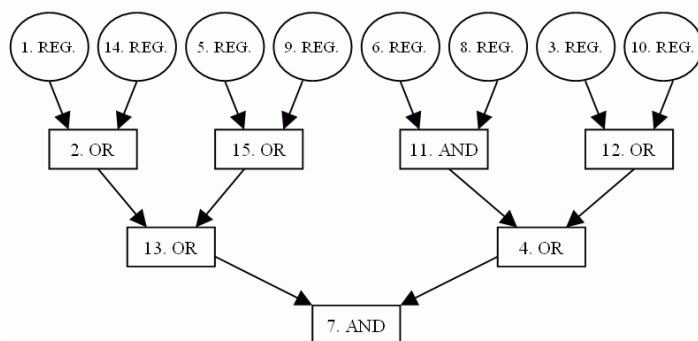
OR T2,T3,T4;

AND T6,R6,R8;

OR T7,R3,R10;

OR T5,T6,T7;

AND T1,T2,T5;



Příklad sítě typu diamant, vstup

Výstup