**Задача B1. K-периферия**

**Автор: Руско Шиков**

Даден е неориентиран свързан граф с *N* върха и *M* ребра. Върховете са номерирани с числата от 1 до *N*. Между всеки два върха има най-много едно ребро. Дължината на всяко ребро е равна на 1. Разстояние между два върха се определя като дължината на най-късия път между тях. Зададен е конкретен връх с номер *C* и цяло, положително число *K*. *K*-периферия на графа спрямо върха с номер *C* се нарича множеството от всички върхове, които се намират на разстояние **по-голямо** от *K* от върха с номер *C.*

Напишете програма **periphery**, която, по зададен граф, връх *C* и число *K*, намира и извежда върховете от *K*-периферията на графа.

**Вход.** От първия ред на стандартния вход се въвеждат четири цели положителни числа *N*, *M*, *C* и *K*, разделени с по един интервал. Следват *M* реда, задаващи всички ребра на графа. Всеки ред съдържа две различни цели положителни числа *u* и *v*, разделени с интервал – номерата на върхове, между които съществува ребро.

**Изход.** На първия ред на стандартния изход програмата трябва да изведе едно цяло число *P –* броя на върховете в *K*-периферията. Ако *P*>0, на всеки от следващите *P* реда програмата трябва да изведе по едно цяло положително число – поредния номер на връх от *K*-периферията. **Върховете трябва да се извеждат в нарастващ ред на номерата им**.

**Ограничения.** 0 < *N, M* ≤ 100 000, 0 ≤ *K < N*, 1 ≤ *C, u, v* ≤ *N*

|  |  |
| --- | --- |
| **Пример 1**  **Вход**  7 8 2 1  1 2  1 3  1 4  1 5  2 3  2 6  2 7  3 6  **Изход**  2  4  5 | **Пример 2**  **Вход**  7 8 1 2  1 2  1 3  1 4  1 5  2 3  2 6  2 7  3 6  **Изход**  0 |