Задача 1. По даден ориентиран непретеглен граф:

- да се провери има ли примки в графа
- да се провери дали графът е пълен (всеки връх е свързан с всички останали)
- да се провери дали графът всъщност е неориентиран (т.е. има ребро между А и Б тогава и само тогава, когато има ребро между Б и А)

Задача 2. В курса Програмиране за напреднали са записани N студенти. Една част от тях посещават курса редовно и се познават помежду си, докато друга присъстват само на контролните работи през семестъра. Малко преди началото на сесията, в края на една от последните лекции, лекторът споделя, че защитата на курсовите проекти ще се проведе малко преди официално опоменатата дата, а също така и че мястото на защитата ще бъде различно от опоменатото.

С настъпването на сесията, лекторът обявява в Мудъл датата на защитата, но пропуска да посочи мястото. Студентите, които са присъствали на гореспоменатата лекция знаят, но тези, които са отсътвали, не.

Като се има предвид познанствата между студентите, както и че е известно, кой има пълната информация, да се напише програма, която отговаря на следните въпроси:

- Дали даден студент А може да разбере правилната локация за защитата.
- Дали всички студенти, които предстои да се явят на защита, могат да разберат правилната локация.
- Да се определи степента на доверие на даден студент. Степен на доверие на студент А наричаме минималният брой студенти, които трябва да знаят необходимата информация, така че тя да може да достигне до студент А.

ПС. Считаме, че ако двама студенти се познават, то те обменят информация.

Задача 3. Даден е ориентиран граф, във върховете на който са разположени букви. Да се определи дали дадена дума може да се прочете след последователно преминаване през върховете на графа.

Задача 4. Нека за неориентиран граф G имаме n на брой върхове, зададени чрез техния номер, и m на брой ребра. Освен това е даден масив topValues с n елемента, в който за всеки връх се пази съответна стойност, с която този връх може да се асоциира, т.е. в topValues[i] се пази стойност, която съответства на i+1-вия връх в графа. Масивът topValues е независим от представянето и дефинирането на графа.

Реализирайте функция, която намира сумата от стойностите на онези върхове в G, които се явяват минимални в рамките на компонентата за свързаност, на която принадлежат.

(Заб.: ако един връх не е свързан с нито един друг връх, то той се брои като самостоятелна компонента на свързаност.)

Пример:

В графа G има върхове от 1 до 10

m = 5 - пет ребра, зададени чрез двойка върхове:

- *1 2*
- *3 4*
- *5 6*
- *7 8*
- *9 10*

topValues[] = {10, 60, 20, 70, 30, 80, 40, 90, 50, 100}

Изход: 150

Задача 5. Даден е списък с хора. Всеки човек има име и възраст. Да се напише програма, която извежда всички двойки хора на екрана и дава възможност на потребителя да опише връзката между тях (приятели, колеги, без връзка). Така получените данни да се съхранят в неориентиран граф. От получените данни да се изведат следните справки:

- човек с най-много приятели
- човек с най-малко приятели
- човек с най-много колеги
- човек с най-малко колеги
- двамата души, които са свързани индиректно с най-много хора между тях

Задача 6. Разглеждаме ориентиран граф. Двупосочен път в ориентиран граф наричаме път, в който всяко ребро има обратно, т.е. огледалният образ на пътя също е път в графа. Да се напише булева функция hasTwoWayPath, която по подадени начален връх start и дължина length проверява, дали в графа съществува ацикличен път с дължина поне length, започващ от върха start, който е двупосочен.

Задача 7. Да се напише функция bool isEulerian([подходящ тип] graph), която по подаден ориентиран претеглен граф, представен чрез матрица на съседство, проверява дали графът е Ойлеров или не.