



Софийски университет „Св. Климент Охридски“  
Факултет по математика и информатика

# ТЕМА ЗА ПРОЕКТ

към курс „Структури от данни и програмиране“

## Карта на град

Да се реализира програма, която представя карта на град. Информацията за картата се съдържа в текстов файл. На всеки ред във файла на първа позиция стои името на дадено кръстовище, а след него двойки от имена на други кръстовища и дължината на улицата от първото кръстовище до второто например:

**Попа БСФС 150 5Къошета 1000**

**БСФС Попа 150**

**5Къошета НДК 400**

**НДК Попа 800**

Забележете, че е възможно да има еднопосочни улици, също така е възможно улиците в двете различни посоки да са различни по дължина, но не е възможно да има две различни улици в една и съща посока между едни и същи кръстовища.

За тази карта да се реализират:

1. Проверка дали има път между две зададени кръстовища.
2. Намиране на трите най-кратки пътя между две зададени кръстовища.
3. При наличието на затворени кръстовища (които да бъдат подавани като списък) да се намери трите алтернативни най-кратки пътища между две зададени кръстовища.
4. Проверка дали при дадено кръстовище за начална точка е възможно да обиколим част от града и върнем обратно там, откъдето сме тръгнали.
5. Проверка дали можем да направим пълна туристическа обиколка на всички улици без да минаваме по една и съща улица два пъти. Ако такава обиколка е възможна, да се намери маршрута на тази обиколка.
6. Проверка дали е възможно да стигнем от дадено кръстовище до всички останали.

7. Намиране на всички задънени улици (еднопосочна улица която свършва в кръстовище от което не излизат улици). Улиците описваме като двойка от имената на кръстовището от което започва и кръстовището, където свършва.

Бонус: Да се реализира интерактивен режим на изпълнение на програмата. Тя се стартира в този режим като при изпълнение на програмата се даде първи параметър -i, последван от файла с картата и кръстовище на което се намираме. Чрез конкретния режим се дава достъп до следните команди:

1. **location** — да се извежда кръстовището на което се намираме в момента
2. **change @** — да променя кръстовището на което се намираме
3. **neighbours** — да извежда всички съседни кръстовища на настоящото (съседно кръстовище е такова, което можем да достигнем от настоящото, без да преминаваме пред междинни кръстовища)
4. **move @** — да премества текущата позиция от кръстовището, на което се намираме, до кръстовището, посочено на мястото на @, и извежда последователността от кръстовища през които се преминава, за да се достигне до там. Ако път от нашето кръстовище до подаденото няма, информация за това да се изведе на екрана и настоящото кръстовище да не бъде променяно
5. **close @** — да се добави кръстовището, посочено в @, към списъка на затворените кръстовища
6. **open @** — премахва кръстовището от списъка @ от списъка на затворените кръстовища
7. **closed** — извежда всички затворени кръстовища
8. **tour** — извежда маршрут на туристическа обиколка на града