

Задачи:

1. Да се напише темплейтна функция, която приема като аргументи 2 сортирани опашки и като резултат връща отново сортирана опашка, с елементите от първоначалните 2.
2. Реализирайте queue с 2 stack-а. ☺
3. Ванката отново е на яко приключение - този път е попаднал в Матрицата. Намира се в квадратно поле с размери $n \times n$. В нея някои клетки са свободни (.), а други блокирани - (X). Знаете, че Ванката се намира на стартова позиция - (start_x, start_y). Помогнете на Ванката да стигне до позиция (destination_x, destination_y), като от вас се иска да кажете на Ванката за **КОЛКО ХОДА** ще стигне до destination позицията.

PS. Тъй като Ванката е много як, може да се движи по цял ред или колона, стига да няма (X), който да го блокира. Тогава той спира на предходната клетка.

Пример: Ванката е в следната матрица с размери 3x3:

```
...  
.X.  
...
```

Стартовата му позиция е (0, 0). Значи се намира в горният ляв ъгъл. Позицията, до която иска да стигне е (1, 2). Тоест пътят му е (0, 0) -> (0, 2) -> (1, 2). Това отнема 2 хода.

Вход

На първият ред се вкарва числото **n** - размерността на матрицата. На следващите **n** реда се въвежда самата матрица - като за свободна клетка се използва (.), а за блокирана - (X).

На последните 2 реда се задават двойките (start_x, start_y) и (destination_x, destination_y)

Изход

Минималният брой ходове, за които Ванката ще стигне до желаната позиция. Ако позицията е недостижима нека функцията връща -1.

4. Да се създаде приоритетна опашка с 3 нива на приоритет – **LOW**, **NORMAL** и **HIGH**. При добавяне на елемент в опашката може да се посочи приоритета. Когато не се посочи приоритет, по **default** приоритета е **NORMAL**. При извикване на функция pop(), първо се извеждат елементите с по-висок приоритет, т.е. HIGH и ако няма елементи там, тогава NORMAL и накрая LOW. Нека се реализират следните методи:
 - push()
 - pop()
 - print()
 - front()
 - isEmpty()