

# 1. Manhattan

Программа принимает на вход координаты улиц города, у которого все кварталы прямоугольные. В некоторых кварталах (а может даже и на дорогах или... перекрестках) этого странного города расположены школы. Программа должна по введенным координатам школьника узнать: какая школа для него ближайшая. Этот грустный город устроен так, что в нем есть несколько правил для школьников:

- 1) Школьники могут ходить только по прямым, параллельным улицам города.
- 2) Школы все разные и каждый школьник, при возможности, идет в школу с наименьшим номером советского квартала.
- 3) Если в районе школьника есть школа, то его родители считают, что ему незачем переходить лишний раз дорогу и отправляют его туда.
- 4) Переходить дорогу можно только на перекрестке(и дети, живущие на дорогах, в школы не ходят!)

**Входные данные:** Сначала вводится количество "вертикальных" улиц потом - количество горизонтальных. После этого в отдельных строках вводятся абсциссы вертикальных улиц, потом - ординаты горизонтальных. Потом вводится количество школ и, по мере требования программы, их координаты в отдельных строках: сначала  $x$ , затем  $y$ .

**Входные-Выходные данные:** После всей подготовительной работы, по запросу программы, вводится количество горожан. После этого в отдельных строках вводятся координаты первого гражданина, после чего сразу выводится номер школы, ему подходящий. Потом координаты второго гражданина - выводится его школа, и т.д.

**Алгоритм:** Итак, как работает программа. По окончании ввода координат улиц, они нумеруются в порядке возрастания (горизонтальные и вертикальные улицы нумеруются независимо). После чего следует ввод координат школ, параллельно с которым производится определение районов, в которых находятся школы (здесь нам помогает функция "сортировка по показывающей", между какими улицами расположена школа с введенными координатами). Далее, для каждого перекрестка перебором находим ближайшую школу, на чем подготовительные работы и заканчиваются.

Затем идет ввод координат горожан. Для каждого горожанина сначала проверяем: есть ли в его квартале школа (следует отметить, что кварталы, ограниченные лишь двумя или тремя улицами мы тоже считаем кварталами). Если есть одна или несколько - выводим минимальный номер школы в этом районе. Если в районе школ нет, то смотрим на школы, ближайшие к перекресткам, являющимися вершинами данного квартала(их может быть 1, 2 или 4). И, путем перебора, выбираем ближайшую школу из рассматриваемых.

**Вывод:** Подготовительная часть программы работает за  $O(n^3)$ , где  $n = \max(\text{количество школ, количество горизонтальных дорог, количество вертикальных дорог})$ . При этом,

в дальнейшем при вводе координат горожанина она за  $O(1)$  выдает номер ближайшей школы. Таким образом, пожертвовав довольно длительное время на обработку информации, мы сможем быстро выводить искомые ответы много-много раз. Что же касается памяти, то программа хранит массивы данных размерами  $X*Y$ ,  $X*Y$  и  $(X + 1)*(Y + 1)$ , где  $X$  и  $Y$  - количество вертикальных и горизонтальных улиц соответственно. Поэтому программа требует довольно много памяти при большом количестве улиц. Однако при наличии необходимого объема памяти программа работает очень хорошо и ею можно пользоваться.