**def resultSaver(fileName, minimumNode):**

fileName fájlban minimumNode helyen módosítja a parkoló foglaltságát emptyről (e) not emptyre (ne)

**def mapBuilder(mapFile):**

mapFile fileból felépíti azt a gráfot, amivel az útvonalkereső tud dolgozni. A szeparátor: tabulátor. Maga a fájl az egy szomszédsági mátrix, amiben minden sor végére oda van illesztve, hogy az adott node parkoló (p) vagy pedig út (r), illetve hogy üres (e) vagy nem üres (ne) az adott node

**def minimumDistance(distance, visitedNode, numberOfNodes):**

végigmegy a gráf egy során, megvizsgálja, hogy a node-on járt-e már, illetve hogy az adott gráfútköltség kisebb-e mint az addig ismert minimum érték, és visszatér a kapott minimum távolság indexével a distance sortömbből

**def pathBuilder(parent, minimumNode, path):**

parent szülőtömbből, minimumNode nodeig felépíti az útvonalat a path tömbbe, és azzal visszatér

**def printSolution(distance, nodeNumber, graph, start, fileName, parent):**

megkeresi a távolságok közül a legkisebbet (-> a start nodehoz legközelebbi üres parkolóhelyet) illetve annak az útvonalnak a költségét, illetve kiíratja a parkoló nodeok állapotát. Ha a következő minimumNode már nagyobb, mint a gráf sorszáma (azaz végigfutottunk a teljes gráfon, amit soronként vizsgálunk), akkor kiírja, hogy nincsen több szabad parkolóhely, egyébként pedig kiírja a starting nodetól mért legkisebb költséget a legközelebbi üres parkolóhelyhez, magát a parkolóhelyet, a pathBuilder metódussal összeépíti az optimális útvonalat, amit utána ki is ír, illetve a végén kiírja, hogy mennyi hely foglalt a rendelkezésre álló parkolók közül.

**def pathfinder(startingNode, fileName):**

kezdő node és a szomszédsági mátrix megadásával elkezdi felépíteni és kiszámolni az útvonalakat. a gráfot a graphBuilder-el készítjük a fileName fileból, a nodeNumber a gráf hossza lesz. A távolságnyilvántartó tömböt inicializálja MAXINT értékkel, kivéve a starting node-ét, azt 0-ra állítja. Nyilvántart egy tömböt arról, hogy az adott sorban érintettünk már egy nodeot vagy sem, illetve inicializálja a szülőket nyilvántartó tömböt -1 értékekkel. Soronként végigszámolja a minimum értékeket, jegyzi a szülőket, és a végén meghívja a saját kiszámolt paramétereivel a printSolution metódust

A programot a pathfinder függvény meghívásával lehet indítani a Janiék esetében a node 0, a Marciék esetében a node 52 a starting node. A Janiék file-ja a map\_state\_ip.txt, a Marcié pedig a map\_state\_ai.txt bár ezeket (elvileg) ők generálják, de nekem kézzel kellett összerakni. A program létre fog hozni egy runLog.log file-t, amiben logolja a program működését