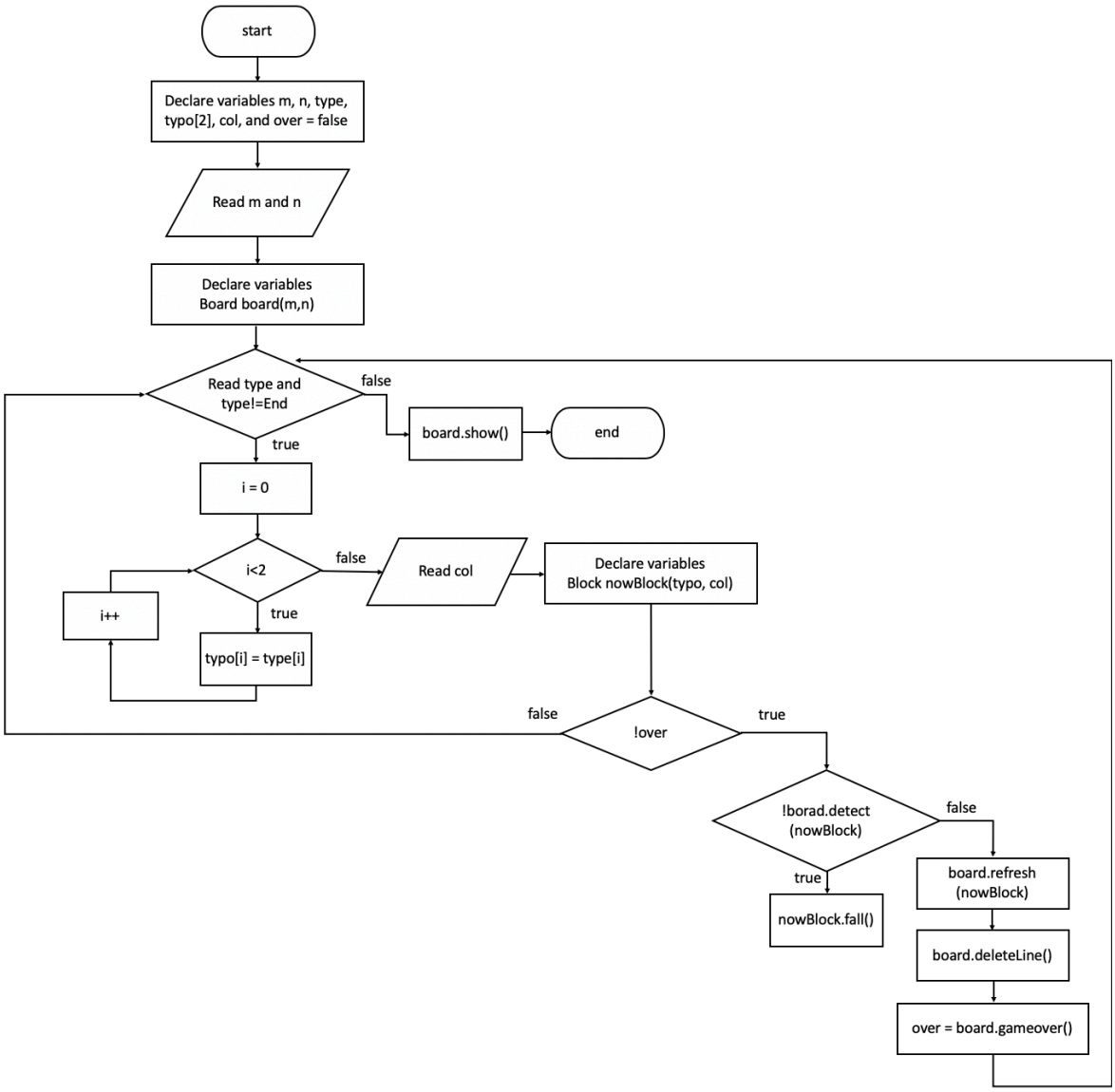
Report

1. Program Flowchart



2. Detail description
   1) 先宣告一個存有所有 block 種類的 array，且多加一個全為 0 的 block
   2) Class Block:
      a. Variables: blockType: 什麼樣子的方塊(英文)，direction: 相同樣式的方塊但經過旋轉(數字)，col: 方塊在哪一行，row: 方塊在哪一列，shape: 方塊的樣子。
      b. Constructor: input 存放 block type 的 array(type[])以及放入的 column(col)，將 row 設成 0，col 設成 input 的 col-1(因為 input 的 column 起始從 1 開始算)，blockType 設成 type[0]，direction 設成 type[1]，若 blockType 是 o 無其他方向，則不設 direction，用 switch

case 將對應的方塊樣式利用 for loop 輸入進 shape。

c. getRow()，getCol()，分別 return 此方塊的 row 以及 col 位子
d. fall()，row += 1，若方塊還可以繼續下降則 call fall function

3) Class Board:
   a. Variables: height: 背景有幾列，width: 背景有幾行，background: 一個 44*15 的 array，高度多了 4 格是 block 準備進入遊戲時放置的位子
   b. Constructor: input 背景的 height(m)以及背景的 width(n)，將 height 設成 m+4，width 設成 n，用 for loop 將 background[m+4][n]全部設成 0
   c. detect(Block &block): 偵測 block 還有沒有下降的空間，設一個 boolean 值 flag 初始值為 false，x 為 block 的 row，y 為 block 的 col，temp 為 x+1，temp 可以偵測下面有沒有空間，用兩個 for loop 掃過 background 在 block 原位子下降一格後的位子能否擺放，若不行則 flag=true，若方塊到底了 flag 也等於 true，最後回傳 flag 值
   d. refreshBlock(Block &block): 將 block 放入對應的 background
   e. deleteLine: 從第 4 列(background 真正開始的位址)開始掃有無需要消掉的列，若有需要消掉則將此列以上的都往下一格，並將 background 的第一列設成 0，重複四次掃描有無需要消除的列，因為一個 block 放入最多可以消除四行
   f. gameOver: 若超出真正 background 的部分(上面四列)有 1 則回傳 true
   g. show: 將 background 印出
4) main: input 一個 m, n，宣告一個 over = false，宣告一個 Board board(m, n)，while 可以 input 近一個 type 且 type 不等於 End，將 type 放入一個 array typo，再 input 此 block 的 col，並宣告一個 Block nowBlock(typo, col)，若 over = false 則 while(board.detect(nowBlock)==false)則 nowBlock.fall()，若 detect==true，則 refreshBlock(nowBlock)將方塊放入該放的位子，並且 deleteLine，最後檢查有沒有 gameOver，最後 board.show()畫出最終的樣子

3. Test case

看 block 能否降落在正確位子上，且是否能一次消除兩行以上的 row