火星（Mars）

衆所周知，最早觸及外太空者，正是諸位法老。最早降落在圖特摩斯一世行星（Thutmus I，現通稱火星）的太空船就是由他們發射。該行星的表面可以看成由正方格組成的 $(2n+1)\times(2n+1)$ 網格，每格或是陸地，或是水域。位於第 i 橫行第 j 豎列（$0\le i,j\le 2\cdot n$）的方格，若其狀態為陸地，則記作 $s[i][j]=$'1'；若為水域，則記作 $s[i][j]=$'0'。

兩格陸地稱為連通，意思是兩者之間可由若干格陸地組成的路徑連接，而路徑上任意連續兩格皆有一條公共邊。行星上的島定義為由兩兩連通的若干格陸地組成的極大集。

當時，太空船的任務是計算行星上島的數目。不過，由於太空船的電腦太古老，此任務並非易事。電腦有記憶體 h，儲存數據的形式是大小為 $(2n+1)\times(2n+1)$ 的二維陣列，其中每個元素可以儲存一個長度不逾 100 的二進制串，其每個字符或為 '0'（ASCII 48），或為 '1'（ASCII 49）。初時，每格記憶體的首位記錄了網格中每格的狀態，$h[i][j][0]=s[i][j]$（對 $0\le i,j\le 2\cdot n$）。h 中其他位的初始值為 '0'（ASCII 48）。

欲處理記憶體的資料，電腦只能存取記憶體某個 3×3 的區塊，然後覆寫該區塊左上角一格的值。以符號表示，電腦可以存取 $h[i..i+2][j..j+2]$（$0\le i,j\le 2\cdot(n-1)$）之值，然後覆寫 $h[i][j]$ 之值。下文稱此過程為**處理第 (i,j) 格**。

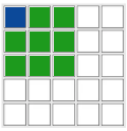
因應電腦有局限，一衆法老籌謀出以下機制：

- 電腦會分 n 輪處理記憶體。
- 第 k $(0\le k\le n-1)$ 輪中，設 $m=2\cdot(n-k-1)$，電腦將對 $0\le i,j\le m$ 處理第 (i,j) 格，次序是按 i 遞增，並對每個 i 按 j 遞增。換言之，電腦處理方格的順序如下：$(0,0),(0,1),\cdots,(0,m),(1,0),(1,1),\cdots,(1,m),\cdots,(m,0),(m,1),\cdots,(m,m)$。
- 末輪（$k=n-1$）中，電腦只處理第 $(0,0)$ 格。此後，$h[0][0]$ 處寫下的值，應等於行星的島數之二進制表示，其中島數的最低位儲存為字串的首個字符。

下圖顯示電腦如何處理 5×5（$n=2$）大小的記憶體。藍色表示即將覆寫的格，有染色表示位於正在處理的子陣列。

第 0 輪期間，電腦依序處理下列子陣列：

第 1 輪期間，電腦只會處理一個子陣列：



給定電腦的運作方式，你的任務是實作一個方法，使該電腦計算出圖特摩斯一世行星上島的數目。

實作需知

你應實作以下子程式：

```
string process(string[][] a, int i, int j, int k, int n)
```

- a：3×3 的陣列，表示要處理的子陣列，具體即 $a = h[i..i+2][j..j+2]$，而 a 的元素皆是長度恰為 100 的字串，每個字符或為 `'0'`（ASCII 48），或為 `'1'`（ASCII 49）。
- i, j：電腦現正處理的方格的行號和列號。
- k：當前是第幾輪。
- n：總輪數，亦表示行星表面大小為 $(2n+1) \times (2n+1)$ 格。
- 此子程式應回傳一個長為 100 的二進制字串。回傳值將記入電腦記憶體的 $h[i][j]$ 處。
- $k = n - 1$ 時，即為最後一次調用。此次調用期間，子程式應回傳行星表面島數之二進制表示，其最低位由指標為 0 的字符（字串的首位）表示，次低位由指標 1 的字符表示，餘可類推。
- 子程式須獨立於任何靜態（static）或全域（global）變量，且其回傳值應只取決於所傳的諸參數。

每筆測資會有 T 個獨立的場景（即不同行星的表面）。你的實作方式，在每個場景的行為，須獨立於各場景的次序，因為在每一場景中，對 `process` 子程式的多次調用，不一定連續發生。不過仍保證在每一場景中，各次調用 `process`，是依照題敍的順序。

此外，在每筆測資中，會有你的程式的若干複本（instances）同時開始運行。記憶和 CPU 時間限制是對全部複本累計而言。任何人若刻意試圖以帶外（out-of-band）方式，在該些複本之間傳遞數據，即視為作弊，將作為取消資格的理由。

具體而言，調用 `process` 子程式期間，儲存到靜態或全域變量的任何資料，不保證在下次調用子程式時仍可取得。

限制

- $1 \le T \le 10$
- $1 \le n \le 20$
- $s[i][j]$ 為 `'0'`(ASCII 48) 或 `'1'`(ASCII 49) (對於所有 $0 \le i, j \le 2 \cdot n$)
- $h[i][j]$ 的長度恰為 100（對 $0 \le i, j \le 2 \cdot n$）
- $h[i][j]$ 的每個字符或是 `'0'`（ASCII 48），或是 `'1'`（ASCII 49）（對 $0 \le i, j \le 2 \cdot n$）

對 `process` 子程式的每次調用：

- $0 \le k \le n - 1$
- $0 \le i, j \le 2 \cdot (n - k - 1)$

子任務

1. (6 分)$n \le 2$
2. (8 分)$n \le 4$
3. (7 分)$n \le 6$
4. (8 分)$n \le 8$
5. (7 分)$n \le 10$
6. (8 分)$n \le 12$
7. (10 分)$n \le 14$
8. (24 分)$n \le 16$
9. (11 分)$n \le 18$
10. (11 分)$n \le 20$

例

例 1

考慮 $n = 1$ 的情況,而 s 如下:

```
1 0 0
1 1 0
0 0 1
```

此例中,行星表面有 3×3 格、2 個島。對 `process` 子程式的調用只有 1 輪。

第 0 輪期間,評測機會調用 `process` 子程式恰好一次:

```
process([["100","000","000"],["100","100","000"],["000","000","100"]],0,0,0,1)
```

留心上式只顯示了 h 的每格的前 3 位。

子程式應回傳 `"0100..."`(略去的位皆為零),因為二進制下 $...0010$ 等於十進制的 2。注意 `...` 取代了省略的 96 個零。

例 2

考慮 $n = 2$ 的情況,而 s 如下:

```
1 1 0 1 1
1 1 0 0 0
1 0 1 1 1
0 1 0 0 0
0 1 1 1 1
```

此例中,行星表面有 5×5 格、4 個島。對 `process` 子程式的調用會有 2 輪。

第 0 輪期間，評測機會調用 `process` 子程式 9 次：

```
process([["100","100","000"],["100","100","000"],["100","000","100"]],0,0,0,2)
process([["100","000","100"],["100","000","000"],["000","100","100"]],0,1,0,2)
process([["000","100","100"],["000","000","000"],["100","100","100"]],0,2,0,2)
process([["100","100","000"],["100","000","100"],["000","100","000"]],1,0,0,2)
process([["100","000","000"],["000","100","100"],["100","000","000"]],1,1,0,2)
process([["000","000","000"],["100","100","100"],["000","000","000"]],1,2,0,2)
process([["100","000","100"],["000","100","000"],["000","100","100"]],2,0,0,2)
process([["000","100","100"],["100","000","000"],["100","100","100"]],2,1,0,2)
process([["100","100","100"],["000","000","000"],["100","100","100"]],2,2,0,2)
```

假設上列調用的回傳值分別為 `"011", "000", "000", "111", "111", "011", "110", "010", "111"`，其中略去的位皆為零。於是，第 0 輪結束時，h 儲存的值會是：

```
"011", "000", "000", "100", "100"
"111", "111", "011", "000", "000"
"110", "010", "111", "100", "100"
"000", "100", "000", "000", "000"
"000", "100", "100", "100", "100"
```

第 1 輪期間，評測機會調用 `process` 子程式恰好一次：

```
process([["011","000","000"],["111","111","011"],["110","010","111"]],0,0,1,2)
```

最終，此子程式應回傳 `"0010000..."`（略去的位皆為零），因為二進制下 $...0000100$ 等於十進制的 4。注意 `...` 取代了省略的 93 個零。

樣例評測程式

樣例評測程式的讀入格式為：

- 第 1 行：T
- 第 i 段（$0 \le i \le T-1$）：表示第 i 場景的一段。
  - 第 1 行：n
  - 第 $2+j$ 行（$0 \le j \le 2 \cdot n$）：$s[j][0]\ s[j][1]\ \ldots\ s[j][2 \cdot n]$

樣例評測程式的輸出格式為：

- 第 $1+i$ 行（$0 \le i \le T-1$）：第 i 場景中，`process` 子程式的最終回傳值，以十進制表示。