

בהנתן מטריצה ריבועית בגודל קבוע $N \times N$ אשר מכילה מספרים שלמים חיוביים בלבד, נגדיר "מסלול" כאוסף של תאים סמוכים. תאים סמוכים יכולים להיות שכנים מימין, משמאל, מלמעלה או מלמטה (לא באלכסון).

עליכם לממש את הפונקציה:

```
bool find_path_sum(int mat[N][N], int sum, int path[N][N]);
```

אשר מקבלת מטריצה ריבועית בגודל קבוע $N \times N$ (ניתן להניח ש N מוגדר ב-`#define`), אשר מכילה מספרים שלמים חיוביים (גדולים ממש מ-0), סכום חיובי `sum`, ומטריצה ריבועית בגודל $N \times N$. הפונקציה צריכה להחזיר `true` במידה וקיים "מסלול" במטריצה `mat` עם הסכום `sum` ו-`false` אחרת. שימו לב שכל תא במסלול נספר פעם אחת.

המטריצה `path`, בגודל $N \times N$, משמשת לסימון מסלול עם הסכום המבוקש: אם נמצא מסלול עם הסכום המבוקש, `path` תכיל את הערך 1 בתאים אשר משתתפים במסלול ואפסים בשאר התאים. אם לא נמצא כזה מסלול `path` צריכה להכיל אפסים בלבד בסיום ריצת הפונקציה. אם קיים יותר ממסלול אחד, `path` תכיל את אחד המסלולים עם הסכום המבוקש. לדוגמא, בהינתן המטריצה `mat` הבאה:

2	41	3	15
1	2	4	6
7	8	10	54
63	22	1	4

והסכום 4, הפונקציה תחזיר `true` והמטריצה `path` יכולה להיות אחת משתי האפשרויות הבאות:

0	0	0	0
0	0	1	0
0	0	0	0
0	0	0	0

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

עבור הסכום 9 לעומת זאת, הפונקציה תחזיר `false` ו-`path` תכיל אפסים בלבד.

עבור הסכום 3, הפונקציה תחזיר `true`, והמטריצה `path` יכולה להיות אחת מ-3 האפשרויות הבאות:

1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0

0	0	0	0
1	1	0	0
0	0	0	0
0	0	0	0

0	0	1	0
0	0	0	0
0	0	0	0
0	0	0	0