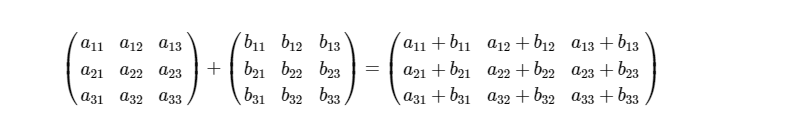
**8.5**(*Algebra: add two matrices*) Write a function to add two matrices a and b and save the result in c.



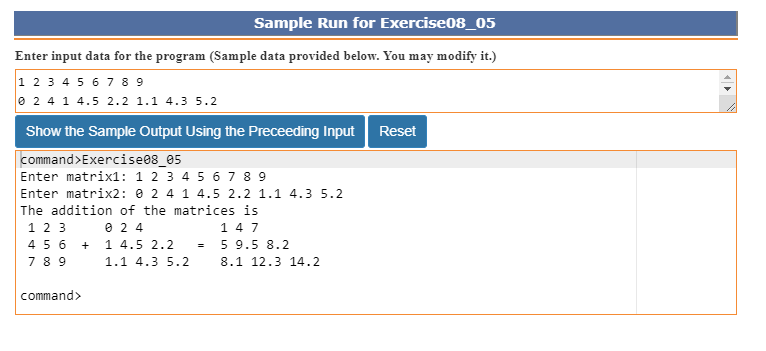
The header of the function is

**const int** N = 3;

**void** addMatrix(**const double** a[][N],

**const double** b[][N], **double** c[][N]) ;

Each element *Cij* is *aij* + *bij*. Write a test program that prompts the user to enter two 3 × 3 matrices and displays their addition. Here is a sample run:



**\*\*8.11**(*Game: nine heads and tails*) Nine coins are placed in a 3 × 3 matrix with some face up and some face down. You can represent the state of the coins using a 3 × 3 matrix with values 0 (head) and 1 (tail). Here are some examples:

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

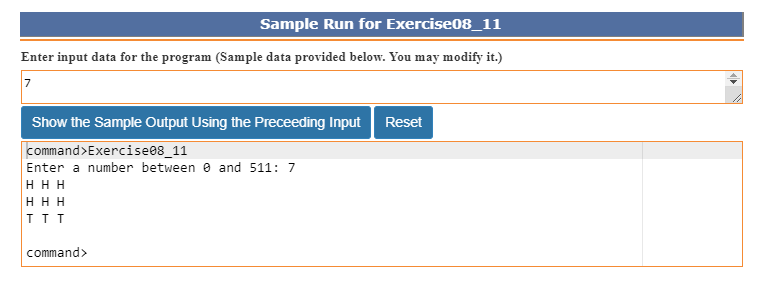
0 1 0 0 0 1 1 0 0 1 1 0 1 1 1

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

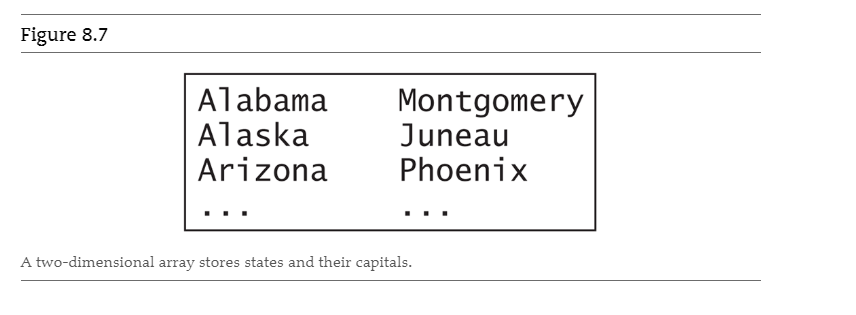
Each state can also be represented using a binary number. For example, the preceding matrices correspond to the numbers

000010000 101001100 110100001 101110100 100111110

The total number of possibilities is 512. So you can use decimal numbers 0, 1, 2, 3, . . . , and 511 to represent all states of the matrix. Write a program that prompts the user to enter a number between 0 and 511 and displays the corresponding matrix with characters H and T. Here is a sample run:



**\*\*8.15**(*Guess the capitals*) Write a program that repeatedly prompts the user to enter a capital for a state. Upon receiving the user input, the program reports whether the answer is correct. Assume that 50 states and their capitals are stored in a two-dimensional array, as shown in [Figure 8.7](https://revel-ise.pearson.com/eps/sanvan/api/item/1bbc97a9-78b2-404d-abb0-959a16e2c819/1/file/liang-rfitpwc-4e_Revel_v5i/OPS/xhtml/ch08_pg0012.xhtml#P7001013097000000000000000004752). The program prompts the user to answer all states’ capitals and displays the total correct count. The user’s answer is not case-sensitive.

****

A two-dimensional array stores states and their capitals.

Here is a sample run:

