|  |
| --- |
| 1. but it’s considered an inferiror technique. Why?   since it can be changed their own type (typedef or #define, whatever it is), latter form can't not follow the left-side operand type. |
| 1. The Q&A section shows how to use a letter as an array subscript. Describe how to use a digits as a subscript.   arr[digits – ‘0’] = 1052; |
| 1. Write a declaration of an array named weekend containing seven bool values. Include an initializer that makes the first and last values true; all other values should be false.   bool weekend[7] = { 1, 0, 0, 0, 0, 0, 1 }; |
| 1. (C99) Repeat Exercise 3, but this time use a designated initializer. Make the initializer as short as possible.   bool weekend[7] = { [0] = true, [6] = true }; |
| 1. Write a program fragment that declares an array named fib\_numbers of length 40 and fiils the array with the first 40 Fibonacci numbers.   int fib\_numbers[40] = { 1, 1 };  for (int i = 2; i < sizeof(fib\_numbers) / sizeof(fib\_numbers[0]); i++)  fib\_numbers[i] = fib\_numbers[i - 1] + fib\_numbers[i - 2]; |
| 1. Here’s what the array might look like, with each row representing one digit. I’ve given you the first row of the initializer; fill in the rest.   const int segments[10][7] = {  {1, 1, 1, 1, 1, 1, 0}, // 0  {0, 1, 1, 0, 0, 0, 0}, // 1  {1, 1, 0, 1, 1, 0, 1}, // 2  {1, 1, 1, 1, 0, 0, 1}, // 3  {0, 1, 1, 0, 0, 1, 1}, // 4  {1, 0, 1, 1, 0, 1, 1}, // 5  {1, 0, 1, 1, 1, 1, 1}, // 6  {1, 1, 1, 0, 0, 0, 0}, // 7  {1, 1, 1, 1, 1, 1, 1}, // 8  {1, 1, 1, 1, 0, 1, 1}, // 9  }; |
| 1. Using the shorcuts described in Section 8.2, shrink the initializer for the segments array (Exercise 6) as much as you can   const int segments[10][7] = {  {1, 1, 1, 1, 1, 1 }, // 0  {0, 1, 1 }, // 1  {1, 1, 0, 1, 1, 0, 1}, // 2  {1, 1, 1, 1, 0, 0, 1}, // 3  {0, 1, 1, 0, 0, 1, 1}, // 4  {1, 0, 1, 1, 0, 1, 1}, // 5  {1, 0, 1, 1, 1, 1, 1}, // 6  {1, 1, 1 }, // 7  {1, 1, 1, 1, 1, 1, 1}, // 8  {1, 1, 1, 1, 0, 1, 1}, // 9  }; |
| 1. Write a declaration for a two-dimensional array named temperature\_readings that stores one month of hourly temperature readings.   float temperature\_readings[30][24]; |
| 1. Using the array of Exercise 8, write a program fragment that computes the average temperature for a month (averaged over all days of the month and all hours of the day.)   double for\_month = 0;  double for\_day = 0;  for (int day = 0; day < 30; day++) {  for\_day = 0.0;  for (int hour = 0; hour < 24; hour++) {  for\_day += temperature\_readings[day][hour];  }  for\_month += for\_day / 24.0;  } |
| 1. Write a declaration for an 8 x 8 char array named chess\_board. Include an initializer that puts the following data into the array (one character per array element):   char chess\_board[8][8] = {  { ‘r’, ‘n’, ‘b’, ‘q’, ‘k’, ‘b’, ‘n’, ‘r’ },  { ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’ },  { ‘ ’, ‘.’, ‘ ‘, ‘.’, ‘ ’, ‘.’, ‘ ’, ‘.’ },  { ‘.‘, ‘ ‘, ‘.’, ‘ ’, ‘.’, ‘ ’, ‘.’, ‘ ’ },  { ‘ ’, ‘.’, ‘ ‘, ‘.’, ‘ ’, ‘.’, ‘ ’, ‘.’ },  { ‘.‘, ‘ ‘, ‘.’, ‘ ’, ‘.’, ‘ ’, ‘.’, ‘ ’ },  { ‘r’, ‘n’, ‘b’, ‘q’, ‘k’, ‘b’, ‘n’, ‘r’ },  { ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’, ‘p’ },  }; |
| 1. Write a program fragment that declares an 8 x 8 char array named checker\_board and then uses a loop to store the following data into the array (one character per array element):   char checker\_board[8][8];  for (int i = 0; i < 8; i++)  for (int j = 0; j < 8; j++)  checker\_board[i][j] = ((i + j) % 2) ? ‘R’ : ‘B’; |