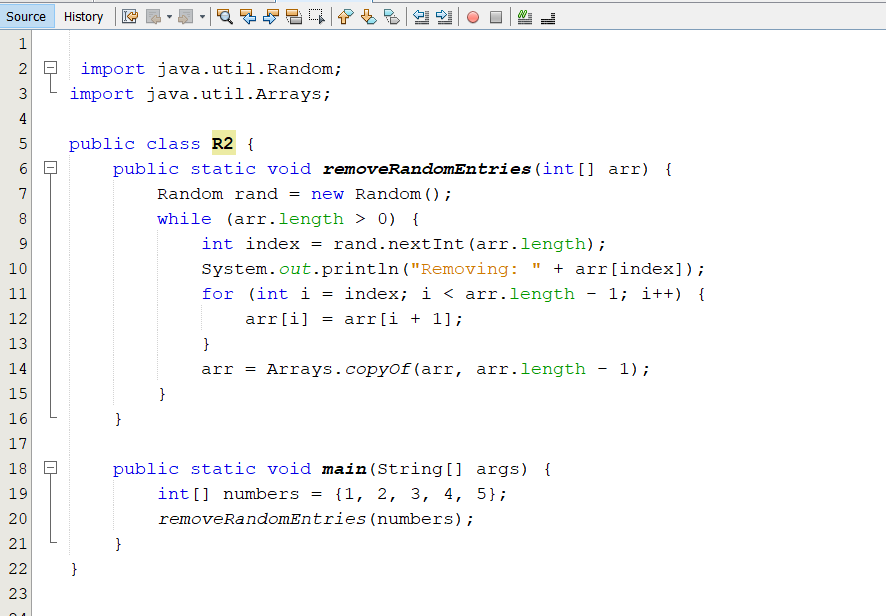
**1 R-3.1 Give the next five pseudorandom numbers generated by the process described on page 113, with a = 12, b = 5, and n = 100, and 92 as the seed for cur.**  
9, 13, 61, 37, 49

**2 R-3.2 Write a Java method that repeatedly selects and removes a random entry from an array until the array holds no more entries.**



**3 R-3.3 Explain the changes that would have to be made to the program of Code Fragment 3.8 so that it could perform the Caesar cipher for messages that are written in an alphabet-based language other than English, such as Greek, Russian, or Hebrew.**  
يجب تعديل الشيفرة بحيث تأخذ في الاعتبار مجموعة الحروف الخاصة باللغة المستهدفة بدلًا من الحروف الإنجليزية A-Z. يمكن تحقيق ذلك باستخدام ترميز Unicode لحروف اللغة المرادة بدلاً من char العادي. أيضًا، يجب تعديل عملية الإزاحة (shift) بحيث تبقى الحروف ضمن نطاق الحروف المستخدمة في اللغة، بدلاً من افتراض وجود 26 حرفًا فقط كما في الإنجليزية.

**4 R-3.4 The TicTacToe class of Code Fragments 3.9 and 3.10 has a flaw, in that it allows a player to place a mark even after the game has already been won by someone. Modify the class so that the putMark method throws an IllegalStateException in that case.**  
يجب تعديل putMark بحيث تتحقق مما إذا كان هناك فائز قبل السماح بوضع علامة جديدة، وإذا كان هناك فائز بالفعل، يتم رمي استثناء IllegalStateException:

public void putMark(int i, int j) {

if (winner() != EMPTY) {

throw new IllegalStateException("Game has already been won!");

}

if (board[i][j] != EMPTY) {

throw new IllegalArgumentException("Cell is already occupied!");

}

board[i][j] = player;

player = -player;

}

**5 R-3.13 What is the difference between a shallow equality test and a deep equality test between two Java arrays, A and B, if they are one-dimensional arrays of type int? What if the arrays are two-dimensional arrays of type int?**

* الاختبار السطحي (shallow equality) يتم باستخدام ==، والذي يتحقق مما إذا كان A وB يشيران إلى نفس كائن المصفوفة في الذاكرة.
* الاختبار العميق (deep equality) يتم باستخدام Arrays.equals(A, B) للمصفوفات ذات البعد الواحد، أو Arrays.deepEquals(A, B) للمصفوفات متعددة الأبعاد.

**6 R-3.14 Give three different examples of a single Java statement that assigns variable, backup, to a new array with copies of all int entries of an existing array, original.**

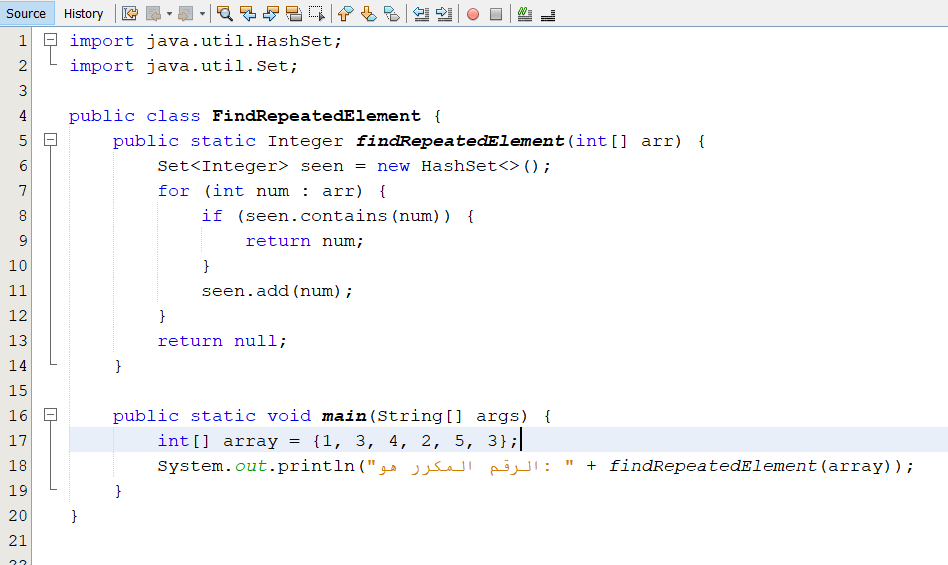
int[] backup = original.clone();

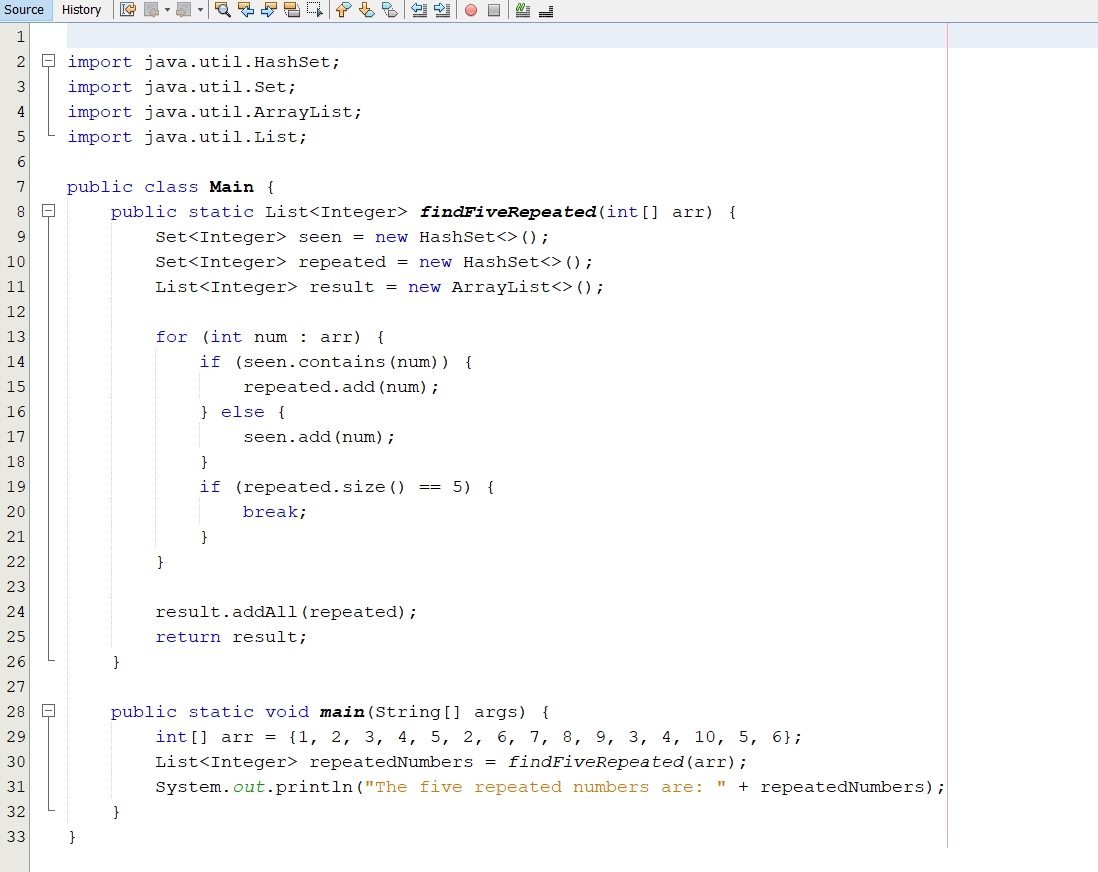
int[] backup = Arrays.copyOf(original, original.length);

int[] backup = new int[original.length];

System.arraycopy(original, 0, backup, 0, original.length);

**7 C-3.17 Let A be an array of size n ≥ 2 containing integers from 1 to n−1 inclusive, one of which is repeated. Describe an algorithm for finding the integer in A that is repeated.**

**8 C-3.18 Let B be an array of size n ≥ 6 containing integers from 1 to n−5 inclusive, five of which are repeated. Describe an algorithm for finding the five integers in B that are repeated.**

**9 C-3.19 Give Java code for performing add(e) and remove(i) methods for the Scoreboard class, as in Code Fragments 3.3 and 3.4, except this time, don’t maintain the game entries in order. Assume that we still need to keep n entries stored in indices 0 to n−1.**

public void add(GameEntry e) {

if (numEntries < board.length) {

board[numEntries++] = e;

}

}

public void remove(int i) {

if (i < numEntries) {

board[i] = board[--numEntries];

board[numEntries] = null;

}

}

**10 C-3.20 Give examples of values for a and b in the pseudorandom generator given on page 113 of this chapter such that the result is not very random looking, for n = 1000.**  
a = 2, b = 0 حيث يؤدي هذا إلى توليد أعداد تتبع نمطًا متكررًا (مثل 0, 2, 4, 6, ...).

**11 C-3.21 Suppose you are given an array, A, containing 100 integers that were generated using the method r.nextInt(10), where r is an object of type java.util.Random. Let x denote the product of the integers in A. There is a single number that x will equal with probability at least 0.99. What is that number and what is a formula describing the probability that x is equal to that number?**  
الرقم هو 0 لأن أي ظهور لـ 0 في المصفوفة سيجعل x = 0.  
الاحتمال يمكن حسابه باستخدام 1 - (9/10)^100 ≈ 0.9999.

**12 C-3.22 Write a method, shuffle(A), that rearranges the elements of array A so that every possible ordering is equally likely.**

public static void shuffle(int[] A) {

Random rand = new Random();

for (int i = A.length - 1; i > 0; i--) {

int j = rand.nextInt(i + 1);

int temp = A[i];

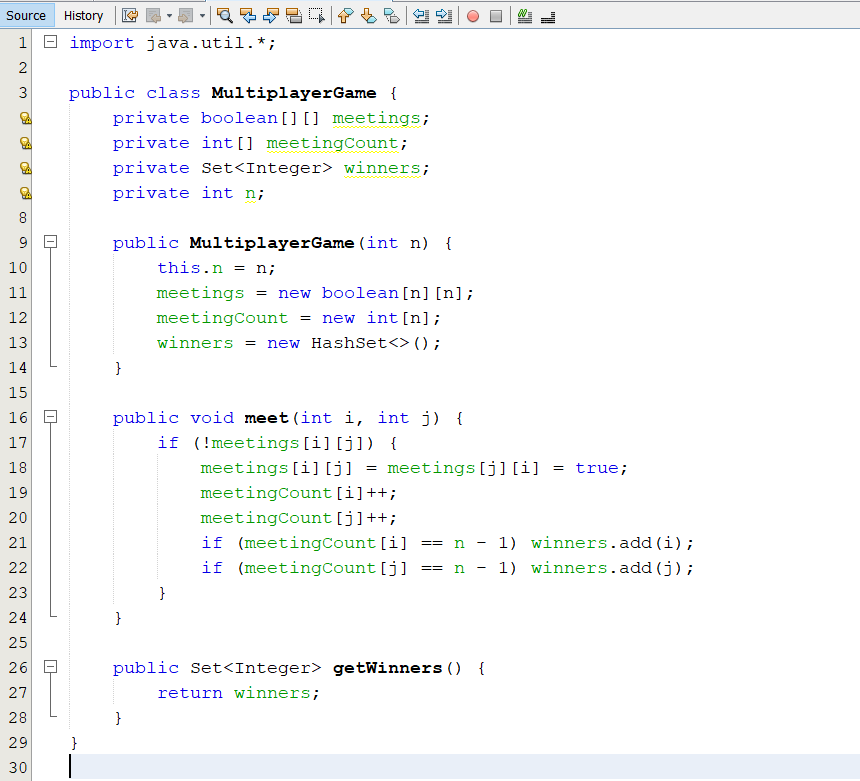
A[i] = A[j];

A[j] = temp;

}

}

**13 C-3.23 Suppose you are designing a multiplayer game that has n ≥ 1000 players, numbered 1 to n, interacting in an enchanted forest. The winner of this game is the first player who can meet all the other players at least once (ties are allowed). Assuming that there is a method meet(i, j), which is called each time a player i meets a player j (with i ≠ j), describe a way to keep track of the pairs of meeting players and who is the winner.**

**14 C-3.24 Write a Java method that takes two three-dimensional integer arrays and adds them componentwise.**

public static int[][][] addThreeDimArrays(int[][][] arr1, int[][][] arr2) {

int x = arr1.length, y = arr1[0].length, z = arr1[0][0].length;

int[][][] result = new int[x][y][z];

for (int i = 0; i < x; i++) {

for (int j = 0; j < y; j++) {

for (int k = 0; k < z; k++) {

result[i][j][k] = arr1[i][j][k] + arr2[i][j][k];

}

}

}

return result;

}