Brandon Kirklen
2015-04-27
CPE 102-22

Home Assignment 4

CPE 102

Chapter 7

20. Section 7.3.6 has two algorithms for removing an element. Which of the two should be used to solve the task described in this section?

Because the order doesn't matter, the correct algorithm would be the first one.

21. It isn't actually necessary to remove the minimum in order to compute the total score. Describe an alternative.

Subtract the minimum after you find the sum.

22. How can you print the number of positive and negative values in a given array, using one or more of the algorithms in Section 6.7?

Count the number of positive values using the matching algorithm then count the negatives.

23. How can you print all positive values in an array, separated by commas?

Just add a print line to the loop of 7.3.4 and include a comma after it.

24. Consider the following algorithm for collecting all matches in an array:

```
int matchesSize = 0;
for (int i = 0; i < values.length; i++)
{
if (values[i] fulfills the condition)
{
matches[matchesSize] = values[i];
matchesSize++;
}
}
```

How can this algorithm help you with Self Check 23?

Use this to find all the positive values then print them using the prior algorithm.

25. Walk through the algorithm that we developed in this section, using two paper clips to indicate the positions for i and j. Explain why there are no bounds errors in the pseudocode.

Brandon Kirklen
2015-04-27
CPE 102-22

Because all the positions possible for I and J are valid positions. There are 8 elements in the array and J is never greater than 8, and I never goes past the fourth object.

26. Take out some coins and simulate the following pseudocode, using two paper clips to indicate the positions for i and j.

```
i = 0
j = size - 1
While (i < j)
Swap elements at positions i and j
i++
j--
```

What does the algorithm do?

It reverses the elements in the array.

27. Consider the task of rearranging all elements in an array so that the even numbers come first. Otherwise, the order doesn't matter. For example, the array

1 4 14 2 1 3 5 6 23

could be rearranged to

4 2 14 6 1 5 3 23 1

Using coins and paperclips, discover an algorithm that solves this task by swapping elements, then describe it in pseudocode.

Walk through the array from the end and beginning. If the end index is even, swap with the first index. Keep going until the indexes meet each other.

28. Discover an algorithm for the task of Self Check 27 that uses removal and insertion of elements instead of swapping.

Walk through the array and remove any odd values then add it to the end of the array. Stop when the first value is equal to the total size of the array.

29. Consider the algorithm in Section 6.7.5 that finds the largest element in a sequence of inputs—not the largest element in an array. Why is this algorithm better visualized by picking playing cards from a deck rather than arranging toy soldiers in a sequence?

Because there is no knowledge of what is coming next in a sequence of inputs where the whole arrangement of soldiers is known when arranging them.

30. What results do you get if you total the columns in our sample data?

The number of gold, silver, and bronze medals given out. 4 each.

Brandon Kirklen
2015-04-27
CPE 102-22

31. Consider an 8 × 8 array for a board game:

```
    int[][] board = new int[8][8];
```

Using two nested loops, initialize the board so that zeroes and ones alternate, as on a checkerboard:

```
    0 1 0 1 0 1 0 1
    1 0 1 0 1 0 1 0
    0 1 0 1 0 1 0 1
    . . .
    1 0 1 0 1 0 1 0
```

Hint: Check whether i + j is even.

```
    For (int x=0; x < 8; x++)
        {
            For (int y=0; y < 8: y++)
            {
                Board[x][y] = (y+x) % 2;
            }
        }
```

32. Declare a two-dimensional array for representing a tic-tac-toe board. The board has three rows and columns and contains strings "x", "o", and " ".

```
        String[][] board = new String[3][3]
```

33. Write an assignment statement to place an "x" in the upper-right corner of the tic-tac-toe board in Self Check 32.

```
        board[0][2] = “x”;
```

34. Which elements are on the diagonal joining the upper-left and the lower-right corners of the tic-tac-toe board in Self Check 32?

```
        board[0][0] and board[1][1] and board[2][2]
```

35. Declare an array list primes of integers that contains the first five prime numbers

```
    (2, 3, 5, 7, and 11).
    ArrayList<Integer> primes =
    new ArrayList<Integer>();
    primes.add(2);
    primes.add(3);
    primes.add(5);
    primes.add(7);
    primes.add(11);
```

36. Given the array list primes declared in Self Check 35, write a loop to print its elements in reverse order, starting with the last element.

```
    for (int I = primes.size()-1; I >=0; i++)
    {
        System.out.println(primes.get(i));
    }
```

37. What does the array list names contain after the following statements?

```
    ArrayList<String> names = new ArrayList<String>;
    names.add("Bob");
    names.add(0, "Ann");
    names.remove(1);
    names.add("Cal");
```

It contains:

“Ann”, “Cal”

38. What is wrong with this code snippet?

```
    ArrayList<String> names;
    names.add(Bob);
```

Names hasn’t been initialized as well as Bob not being a string.

39. Consider this method that appends the elements of one array list to another:

```
    public void append(ArrayList<String> target, ArrayList<String> source)
    {
    for (int i = 0; i < source.size(); i++)
    {
    target.add(source.get(i));
    }
    }
```

What are the contents of names1 and names2 after these statements?

```
    ArrayList<String> names1 = new ArrayList<String>();
    names1.add("Emily");
    names1.add("Bob");
    names1.add("Cindy");
    ArrayList<String> names2 = new ArrayList<String>();
    names2.add("Dave");
    append(names1, names2);
```

names1 contains the names "Emily", "Bob", "Cindy", and "Dave" while names2 contains "Dave".

40. Suppose you want to store the names of the weekdays. Should you use an array list or an array of seven strings?

An array because the names aren’t going to change in number. (Hopefully)

Brandon Kirklen
2015-04-27
CPE 102-22

41. The ch07/section_7 directory of your source code contains an alternate implementation of the problem solution in How To 7.1 on page 334. Compare the array and array list implementations. What is the primary advantage of the latter?

It’s easier to put the inputs into the arraylist.

42. Suppose you modified the code for a method. Why do you want to repeat tests that already passed with the previous version of the code?

Because you might have messed up prior code.

43. Suppose a customer of your program finds an error. What action should you take beyond fixing the error?

Add test cases that make sure all future versions don’t contain the error.

44. Why doesn’t the ScoreTester program contain prompts for the inputs?

Because all the input is from a file rather than from the console.