**Method and array tracing practice**

1. **Arrays.** The following program prints three lines of output. What are the three lines?

**public class Mystery {**

**public static void main(String[] args) {**

**// Part 1**

**int[] a = {0, 1, 2, 3, 4, 5};**

**for (int i = 0; i < 6; i++)**

**System.out.print(a[i/2] + " ");**

**System.out.println();**

**// Part 2**

**for (int i = 5; i >= 0; i--)**

**System.out.print(5 - a[a[i]] + " ");**

**System.out.println();**

**// Part 3**

**for (int i = 0; i < 3; i++) {**

**int t = a[i];**

**a[i] = a[5-i];**

**a[5-i] = t;**

**System.out.print(a[i] + " ");**

**}**

**System.out.println(a[3] + " " + a[4] + " " + a[5]);**

**}**

**}**

1. **Arrays, Functions**

The following two methods do the same job. They each take an ORDERED array of ints

and a target number as arguments.

**public static boolean mystery1(int[] array, int target) {**

**for (int i = 0; i < array.length; i++) {**

**if (array[i] == target)**

**return true;**

**else if (array[i] > target)**

**return false;**

**}**

**return false;**

**}**

**public static boolean mystery2(int[] array, int target) {**

**int low = 0;**

**int high = array.length - 1;**

**while (low <= high) {**

**int mid = (low + high) / 2;**

**if (array[mid] == target)**

**return true;**

**else if (array[mid] < target)**

**low = mid + 1;**

**else**

**high = mid - 1;**

**}**

**return false;**

**}**

Use this array to answer the following questions:

**int[] a = { 2, 5, 11, 14, 15, 27, 31};**

a) What does **mystery1(a, 5)** return?

b) Fill in the trace table to show that **mystery2(a, 5)** returns the same thing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| target | **low** | **high** | **mid** | **return value** |
| 5 | 0 | 6 | 3 |  |
| 5 | 0 | 2 | 1 | true |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

c) What do **mystery1(a, 20)** and **mystery2(a, 20)** return?

d) In general, what do these methods do?

1. **Methods and Input/Output:** Consider the following program:

**1 public class MethodMadness {**

**2 private static int printX( int y ) {**

**3 int x = 2;**

**4 System.out.println( y );**

**5 return 0;**

**6 }**

**7**

**8 private static int printY( int x ) {**

**9 int y = -1;**

**10 System.out.println( x );**

**11 return x;**

**12 }**

**13**

**14 public static void main(String[] args) {**

**15 Scanner sc = new Scanner(System.in);**

**16 int x = sc.nextInt() + 1;**

**16 int y = printY( printX( x ) );**

**17 printX(y);**

**18 }**

* 1. What is printed when we run this program and type in 1 on standard input?
  2. What is printed when we run this program and type in 3 on standard input?

4. **Loops and Arrays:** Consider the following code.

**public class IntegerSort {**

**public static void main(String[] args) {**

**int MAX = 10; // integers are between 0 and MAX-1**

**int[] freq = new int[MAX]; // freq[i] = number of occurrences of i**

**Scanner sc = new Scanner(System.in);**

**for (int i = 0; i < 7; i++){**

**int num = sc.nextInt();**

**freq[num]++;**

**}**

**for (int i = 0; i < MAX; i++) {**

**for (int j = 0; j < freq[i]; j++) {**

**System.out.print(i + " ");**

**}**

**}**

**System.out.println();**

**}**

**}**

Suppose the input to this program is:

**9 2 3 1 0 2 2**

1. Fill in the trace table with the non-zero values of the freq array. Each line of the trace represents one iteration of the while loop above.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. What is printed when we run the remainder of the code with the above input?
2. In general, what do this code do?

Solutions:

1. Arrays. The three parts print out the following three lines:

0 0 1 1 2 2

0 1 2 3 4 5

5 4 3 2 1 0

1. A.True

B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| target | **low** | **high** | **mid** | **return value** |
| 5 | 0 | 6 | 3 |  |
| 5 | 0 | 2 | 1 | true |

C. false

D. Search for a value in a sorted array. They return true, if the value is found; they return false if the value is not found.

1. Methods and I/O

2

0

0

5

0

0

1. Loops and Arrays

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|  |  |  |  |  |  |  |  |  | 1 |
|  |  | 1 |  |  |  |  |  |  | 1 |
|  |  | 1 | 1 |  |  |  |  |  | 1 |
|  | 1 | 1 | 1 |  |  |  |  |  | 1 |
| 1 | 1 | 1 | 1 |  |  |  |  |  | 1 |
| 1 | 1 | 2 | 1 |  |  |  |  |  | 1 |
| 1 | 1 | 3 | 1 |  |  |  |  |  | 1 |

* 1. 0 1 2 2 2 3 9
  2. Print input in a sorted order