

**2017**

**AP Computer Science A Free-Response Questions**

© 2017 The College Board. College Board, Advanced Placement Program, AP, AP Central, and the acorn logo are registered trademarks of the College Board. Visit the College Board on the Web: www.collegeboard.org. AP Central is the offjcial online home for the AP Program: apcentral.collegeboard.org.

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

**COMPUTER SCIENCE A**

**SECTION II**

**Time—1 hour and 30 minutes**

**Number of questions—4**

**Percent of total score—50**

**Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.**

Notes:

 Assume that the interface and classes listed in the Java Quick Reference have been imported where appropriate.

 Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods

are called only when their preconditions are satisfied.

 In writing solutions for each question, you may use any of the accessible methods that are listed in classes

defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**2-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

1. This question involves identifying and processing the digits of a non-negative integer. The declaration of the

Digits class is shown below. You will write the constructor and one method for the Digits class.

public class Digits

{

/\*\* The list of digits from the number used to construct this object.

\* The digits appear in the list in the same order in which they appear in the original number.

\*/

private ArrayList<Integer> digitList;

/\*\* Constructs a Digits object that represents num.

\* **Precondition**: num >= 0

\*/

public Digits(int num)

{ /\* to be implemented in part (a) \*/ }

/\*\* Returns true if the digits in this Digits object are in strictly increasing order;

\* false otherwise.

\*/

public boolean isStrictlyIncreasing() { /\* to be implemented in part (b) \*/ }

}

Part (a) begins on page 4.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**3-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(a) Write the constructor for the Digits class. The constructor initializes and fills digitList with the digits from the non-negative integer num. The elements in digitList must be Integer objects representing single digits, and appear in the same order as the digits in num. Each of the following examples shows the declaration of a Digits object and the contents of digitList as initialized by the constructor.

Example 1

Digits d1 = new Digits(15704);

d1:

0 1 2 3 4

digitList: 1 5 7 0 4

Example 2

Digits d2 = new Digits(0);

d2:

0

digitList: 0

**WRITE YOUR SOLUTION ON THE NEXT PAGE.**

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**4-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

Complete the Digits constructor below.

/\*\* Constructs a Digits object that represents num.

\* **Precondition**: num >= 0

\*/

public Digits(int num)

Part (b) begins on page 6.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**5-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(b) Write the Digits method isStrictlyIncreasing. The method returns true if the elements of

digitList appear in strictly increasing order; otherwise, it returns false. A list is considered strictly increasing if each element after the first is greater than (but not equal to) the preceding element.

The following table shows the results of several calls to isStrictlyIncreasing.

Method call Value returned

|  |  |
| --- | --- |
| new Digits(7).isStrictlyIncreasing() | true |
| new Digits(1356).isStrictlyIncreasing() | true |
| new Digits(1336).isStrictlyIncreasing() | false |
| new Digits(1536).isStrictlyIncreasing() | false |
| new Digits(65310).isStrictlyIncreasing() | false |

**WRITE YOUR SOLUTION ON THE NEXT PAGE.**

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**6-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

Complete method isStrictlyIncreasing below.

/\*\* Returns true if the digits in this Digits object are in strictly increasing order;

\* false otherwise.

\*/

public boolean isStrictlyIncreasing()

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**7-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

2. This question involves the design of a class that will be used to produce practice problems. The following

StudyPractice interface represents practice problems that can be used to study some subject.

public interface StudyPractice

{

/\*\* Returns the current practice problem. \*/ String getProblem();

/\*\* Changes to the next practice problem. \*/ void nextProblem();

}

The MultPractice class is a StudyPractice that produces multiplication practice problems. A MultPractice object is constructed with two integer values: *first integer* and *initial second integer*. The first integer is a value that remains constant and is used as the first integer in every practice problem. The initial second integer is used as the starting value for the second integer in the practice problems. This second value is incremented for each additional practice problem that is produced by the class.

For example, a MultPractice object created with the call new MultPractice(7, 3) would be used to create the practice problems "7 TIMES 3", "7 TIMES 4", "7 TIMES 5", and so on.

In the MultPractice class, the getProblem method returns a string in the format of "*first integer* TIMES *second integer*". The nextProblem method updates the state of the MultPractice object to represent the next practice problem.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**8-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

The following examples illustrate the behavior of the MultPractice class. Each table shows a code

segment and the output that would be produced as the code is executed. **Example 1**

Code segment

StudyPractice p1 = new MultPractice(7, 3); System.out.println(p1.getProblem());

p1.nextProblem();

System.out.println(p1.getProblem());

p1.nextProblem();

System.out.println(p1.getProblem());

p1.nextProblem();

System.out.println(p1.getProblem());

**Example 2**

Code segment

StudyPractice p2 = new MultPractice(4, 12); p2.nextProblem(); System.out.println(p2.getProblem()); System.out.println(p2.getProblem());

p2.nextProblem();

p2.nextProblem();

System.out.println(p2.getProblem());

p2.nextProblem();

System.out.println(p2.getProblem());

Question 2 continues on page 10.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

Output produced

7 TIMES 3

7 TIMES 4

7 TIMES 5

7 TIMES 6

Output produced

4 TIMES 13

4 TIMES 13

4 TIMES 15

4 TIMES 16

**GO ON TO THE NEXT PAGE.**

-**9-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

Interface information for this question public interface StudyPractice String getProblem()

void nextProblem()

Write the complete MultPractice class. Your implementation must be consistent with the specifications and the given examples.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**10-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

3. This question involves analyzing and modifying a string. The following Phrase class maintains a phrase in an instance variable and has methods that access and make changes to the phrase. You will write two methods of the Phrase class.

public class Phrase {

private String currentPhrase;

/\*\* Constructs a new Phrase object. \*/ public Phrase(String p)

{ currentPhrase = p; }

/\*\* Returns the index of the nth occurrence of str in the current phrase;

\* returns -1 if the nth occurrence does not exist.

\* **Precondition:** str.length() > 0 and n > 0 \* **Postcondition:** the current phrase is not modified.

\*/

public int findNthOccurrence(String str, int n) { /\* implementation not shown \*/ }

/\*\* Modifies the current phrase by replacing the nth occurrence of str with repl.

\* If the nth occurrence does not exist, the current phrase is unchanged.

\* **Precondition:** str.length() > 0 and n > 0 \*/

public void replaceNthOccurrence(String str, int n, String repl) { /\* to be implemented in part (a) \*/ }

/\*\* Returns the index of the last occurrence of str in the current phrase;

\* returns -1 if str is not found.

\* **Precondition:** str.length() > 0

\* **Postcondition:** the current phrase is not modified.

\*/

public int findLastOccurrence(String str) { /\* to be implemented in part (b) \*/ }

/\*\* Returns a string containing the current phrase. \*/ public String toString()

{ return currentPhrase; }

}

Part (a) begins on page 12.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**11-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(a) Write the Phrase method replaceNthOccurrence, which will replace the nth occurrence of the string str with the string repl. If the nth occurrence does not exist, currentPhrase remains unchanged.

Several examples of the behavior of the method replaceNthOccurrence are shown below.

Code segments Output produced

Phrase phrase1 = new Phrase("A cat ate late."); phrase1.replaceNthOccurrence("at", 1, "rane"); System.out.println(phrase1);

A crane ate late.

Phrase phrase2 = new Phrase("A cat ate late."); phrase2.replaceNthOccurrence("at", 6, "xx"); System.out.println(phrase2);

A cat ate late.

Phrase phrase3 = new Phrase("A cat ate late."); phrase3.replaceNthOccurrence("bat", 2, "xx"); System.out.println(phrase3);

A cat ate late.

Phrase phrase4 = new Phrase("aaaa"); phrase4.replaceNthOccurrence("aa", 1, "xx"); System.out.println(phrase4);

xxaa

Phrase phrase5 = new Phrase("aaaa"); phrase5.replaceNthOccurrence("aa", 2, "bbb"); System.out.println(phrase5);

abbba

Class information for this question public class Phrase

private String currentPhrase public Phrase(String p)

public int findNthOccurrence(String str, int n)

public void replaceNthOccurrence(String str, int n, String repl) public int findLastOccurrence(String str)

public String toString()

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**12-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

The Phrase class includes the method findNthOccurrence, which returns the nth occurrence of a given string. You must use findNthOccurrence appropriately to receive full credit.

Complete method replaceNthOccurrence below.

/\*\* Modifies the current phrase by replacing the nth occurrence of str with repl.

\* If the nth occurrence does not exist, the current phrase is unchanged.

\* **Precondition:** str.length() > 0 and n > 0 \*/

public void replaceNthOccurrence(String str, int n, String repl)

Part (b) begins on page 14.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**13-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(b) Write the Phrase method findLastOccurrence. This method finds and returns the index of the last occurrence of a given string in currentPhrase. If the given string is not found, -1 is returned. The following tables show several examples of the behavior of the method findLastOccurrence.

Phrase phrase1 = new Phrase("A cat ate late.");

Method call Value returned

|  |  |
| --- | --- |
| phrase1.findLastOccurrence("at") | 11 |
| phrase1.findLastOccurrence("cat") | 2 |
| phrase1.findLastOccurrence("bat") | -1 |

Class information for this question public class Phrase

private String currentPhrase public Phrase(String p)

public int findNthOccurrence(String str, int n)

public void replaceNthOccurrence(String str, int n, String repl) public int findLastOccurrence(String str)

public String toString()

**WRITE YOUR SOLUTION ON THE NEXT PAGE.**

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**14-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

You must use findNthOccurrence appropriately to receive full credit. Complete method findLastOccurrence below.

/\*\* Returns the index of the last occurrence of str in the current phrase;

\* returns -1 if str is not found.

\* **Precondition:** str.length() > 0

\* **Postcondition:** the current phrase is not modified.

\*/

public int findLastOccurrence(String str)

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**15-**

|  |  |  |  |
| --- | --- | --- | --- |
| 15 | 5 | 9 | 10 |
| 12 | 16 | 11 | 6 |
| 14 | 8 | 13 | 7 |

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

4. This question involves reasoning about a two-dimensional (2D) array of integers. You will write two static methods, both of which are in a single enclosing class named Successors (not shown). These methods process a 2D integer array that contains consecutive values. Each of these integers may be in any position

in the 2D integer array. For example, the following 2D integer array with 3 rows and 4 columns contains the integers 5 through 16, inclusive.

2D Integer Array

0 1 2 3

0 1 2

The following Position class is used to represent positions in the integer array. The notation (r,c) will be used to refer to a Position object with row r and column c.

public class Position

{

/\*\* Constructs a Position object with row r and column c. \*/

public Position(int r, int c)

{ /\* implementation not shown \*/ }

// There may be instance variables, constructors, and methods that are not shown.

}

(a) Write a static method findPosition that takes an integer value and a 2D integer array and

returns the position of the integer in the given 2D integer array. If the integer is not an element of the 2D integer array, the method returns null.

For example, assume that array arr is the 2D integer array shown at the beginning of the question.

The call findPosition(8, arr) would return the Position object (2,1) because the

value 8 appears in arr at row 2 and column 1.

 The call findPosition(17, arr) would return null because the value 17 does not

appear in arr.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**16-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

Complete method findPosition below.

/\*\* Returns the position of num in intArr;

\* returns null if no such element exists in intArr.

\* **Precondition**: intArr contains at least one row.

\*/

public static Position findPosition(int num, int[][] intArr)

Part (b) begins on page 18.

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**17-**

|  |  |  |  |
| --- | --- | --- | --- |
| 15 | 5 | 9 | 10 |
| 12 | 16 | 11 | 6 |
| 14 | 8 | 13 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
| (1,1) | (1,3) | (0,3) | (1,2) |
| (2,2) | null | (1,0) | (2,3) |
| (0,0) | (0,2) | (2,0) | (2,1) |

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

(b) Write a static method getSuccessorArray that returns a 2D successor array of positions created

from a given 2D integer array.

The *successor* of an integer value is the integer that is one greater than that value. For example,

the successor of 8 is 9. A 2D *successor array* shows the position of the successor of each element in a given 2D integer array. The 2D successor array has the same dimensions as the given 2D integer array.

Each element in the 2D successor array is the position (row, column) of the corresponding 2D integer array element’s successor. The largest element in the 2D integer array does not have a successor in the

2D integer array, so its corresponding position in the 2D successor array is null.

The following diagram shows a 2D integer array and its corresponding 2D successor array. To illustrate the successor relationship, the values 8 and 9 in the 2D integer array are shaded. In the 2D successor array, the shaded element shows that the position of the successor of 8 is (0,2) in the 2D integer array. The largest value in the 2D integer array is 16, so its corresponding element in the 2D successor array is null.

2D Integer Array 2D Successor Array

0 1 2 3 0 1 2 3

0 1 2

0 1 2

Class information for this question public class Position

public Position(int r, int c)

public class Successors

public static Position findPosition(int num, int[][] intArr) public static Position[][] getSuccessorArray(int[][] intArr)

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

**GO ON TO THE NEXT PAGE.** -**18-**

2017 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

Assume that findPosition works as specified, regardless of what you wrote in part (a). You must use findPosition appropriately to receive full credit.

Complete method getSuccessorArray below.

/\*\* Returns a 2D successor array as described in part (b) constructed from intArr.

\* **Precondition**: intArr contains at least one row and contains consecutive values.

\* Each of these integers may be in any position in the 2D array.

\*/

public static Position[][] getSuccessorArray(int[][] intArr)

**STOP**

**END OF EXAM**

© 2017 The College Board.

Visit the College Board on the Web: www.collegeboard.org.

-**19-**