|  |  |  |
| --- | --- | --- |
| 1. Refer to the code below to answer the following   String s = “Get here Thanksgiving!”;  String m = “er”;  int j = 8, z = 99; | | |
| (a)  int k = s.indexOf(m);  System.out.println(k); |  | |
| (b)  int k = s.indexOf(‘T’);  System.out.println(k); |  | |
| (c)  char p = s.charAt(6);  System.out.println(p); |  | |
| (d)  int k = s.indexOf(z);  System.out.println(k); |  | |
| (e)  int k = s.indexOf(‘g’, j);  System.out.println(k); |  | |
| (f)  char p = s.charAt(z – 90);  System.out.println(p); |  | |
| (g)  int k = s.indexOf(m, 15);  System.out.println(k); |  | |
| (h)  int k = s.indexOf(z + 2, 4);  System.out.println(k); |  | |
| (i)  boolean k = s.contains(m);  System.out.println(k); |  | |
| (j)  String s2 = “ JAVA “;  String k = “!” + s2.trim() + “!”  System.out.println(k); |  | |
| (k)  System.out.println(m.compareTo(s)); |  | |
|  | | /11 |

|  |  |
| --- | --- |
| 1. The Alphabetize class below, alphabetizes three words. Consider the following examples. Write the Alphabetize class. | |
| |  |  | | --- | --- | | **Values of Strings s1, s2, and s3 before** | **Values of s1, s2, and s3 after** | | String s1 = “cat”;  String s2 = “car”;  String s3 = “dog”; | String s1 = “car”;  String s2 = “cat”;  String s3 = “dog”; | | String s1 = “dog”;  String s2 = “cat”;  String s3 = “car”; | String s1 = “car”;  String s2 = “cat”;  String s3 = “dog”; | | |
| public class Alphabetize{  public static void main(String args[]){      }  } | |
|  | /5 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Write an algorithm that could be used to count the number of times a string occurs in another string. Consider the examples below[[1]](#footnote-1). This algorithm requires that you incorporate a loop along with the *substring()* and *length()* methods. Feel free to use other methods as necessary.  |  |  |  | | --- | --- | --- | | **String to search** | **String to find** | **Occurrences** | | BAAB | AA | 1 | | AAAAA | AA | 2 | | AABABABAA | ABA | 2 | | ABBAABB | ABA | 0 | | |
| public class FindOccur{  public static void main(String args[]){    }  } | |
|  | /5 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. The *removeSub* class below removes string from another string and composes the result. If the substring is not found the original string is returned. Consider the examples below,  |  |  |  | | --- | --- | --- | | **String to search** | **Substring to remove** | **Result** | | ABAAAAABAAAAABA | ABA | AAAAAA | | ABAAAAABAAAAABA | ABAA | AAAAABA | | ABAAAAABAAAAABA | AABAA | ABAAAAAABA | | ABAAAAABAAAAABA | ACA | ABAAAAABAAAAABA |   You must use the scanner methods *setDelimiter()* and *hasNext()* in your solution. | |
| public class removeSub{  public static void main(String args[]){  Scanner sc = new Scanner("ABAAAAABAAAAABA");  String delimiter = /\*some substring to be removed\*/  }  } | |
|  | /3 |

1. Adapted from the 2020 AP Computer Science A Exam [↑](#footnote-ref-1)