1 What is the maximum number of recursive calls to `BinarySearch()` that will be made on an array of $20,000$ elements? (3)

2 In your own words, explain why data should be sorted prior to a call to `BinarySearch()`. What might happen if the data is not sorted? (3)

3 Explain why it is not generally advisable to check to see if data is sorted before employing binary search techniques on it. How might you detect unsorted data within the binary search method? (5)

4 Overload `BinarySearch()` to accept an array of any `Comparable` data type. (5)

5 Implement the `closestSum()` method. This method should take as parameters an array of integers, `a`, and an integer, `sum`, and output the two elements of `a` whose sum is *closest* to `sum`. (10)
Note: You may assume that `a` is in sorted order.

6 *Wordsearch Solver.* Implement the `findWord()` method. This method should take as parameters a two-dimensional array of `char` values and a single `String` and output the location and direction (i.e., "Diagonal, Down-Right") of the word within the array. "Word Not Found" should be printed if the given word is not in the puzzle. See below for an example. (20)

Example: Consider the following array of characters, stored in variable `puzzle`.

O	E	Y	O	K	E	R	A
E	R	N	B	R	E	N	N
L	S	S	E	O	W	Y	R
B	U	D	V	D	D	K	B
R							
O	U	N	E	O	T	Y	S
W	E	O	L	E	O	N	S
S	O	T	K	R	O	R	P

A call to `findWord(puzzle, "MONITOR")` should output: `(5, 2) Horizontal, Right`.
A call to `findWord(puzzle, "COMPUTER")` should output: `Word Not Found`.

Note: You should assume the word can be arranged within the puzzle in any direction, including backwards!