# IB/AP Computer Science 25 Basic Algorithms

|  |  |  |  |
| --- | --- | --- | --- |
|  | Problem/Class Name | MethodName | Description |
| 1 | AverageValue | public double calculateAverage() | Average Value returns the average value of an array of doubles. |
| 2 | RangeValue | public double calculateRange() | Range Value takes an array of double and returns the range (max - min) of the values  in the array |
| 3 | MaximumPosition | public int calculateMaximumPosition() | Maximum position takes an array of double as an input and returns the index of the maximum  value in the array. |
| 4 | PartyGreeting | public String calculateGreeting() | Party Greeting takes an array of String containing names and constructs an appropriate greeting  starting with Hello, ending a !, and with commas between the names. |
| 5 | SeasonCalendar | public String calcSeason() | Season calendar takes a month and day as a date and returns a season as follows Dec 21 - Mar 20: Winter; Mar 21 - Jun 20: Spring; Jun 21 - Sep 20: Summer; Sep 21 - Dec 20: Autumn. You can handle month/day combinations that are invalid as you wish. |
| 6 | SumOfDigits | public int calculateSumOfDigits() | Sum of digits takes an integer and returns the sum of the digits in the integer. If integer is negative, then Sum of digits will return the sum of the digits negated. |
| 7 | NthFibbonacci | public int calcNthFibbonacci() | Nth Fibonacci calculates the nth Fibonacci number. Apologies for misspelling “Fibonacci” in the code and elsewhere. |
| 8 | LastFirstReversal | public String reverseFirstLast() | Last First Reversal takes a name in the format "Last, First M" and returns a name String in the format "First M Last". If the comma either does not exist or is the first or last character in the original  name string, you should return the original name string. |
| 9 | CalculateHeronArea | public double calculateHeronArea() | Triangle Area Heron takes the length of three sides of a triangle and uses Heron's formula to calculate the area. If any of the lengths are 0 of negative or if the three sides would not make a valid triangle  0.0 should be returned. Herons formula is the area of triangle is:  square root of (s \* (s - side1) \* (s - side2) \* (s - side 3)) where s = (side1 + side2 + side3)/2 |
| 10 | IsIncreasingIntegerArray | public boolean isIncreasing() | Is Increasing Integer Array takes an array of integers and returns true if each element is greater  than or equal to the previous element, false otherwise. |
| 11 | FindString | public int findLocation() | Find String determines whether one String exists within another. If the String to be found is within the target String, Find String will return the starting index in the target string where the match starts. Otherwise, it returns -1. Find String is case sensitive. You may not use the String indexOf() method in your solution. |
| 12 | StackingBlocks | public String drawStackingBlocks() | Stacking blocks takes an integer parameter and draws a series of blocks stacked upon each other based upon the parameter. You may not use the String .repeat() method in your solution. See example output at the end of this document. |
| 13 | To2DArray | public int[][] calc2DArray() | To 2D Array takes a 1D array of integers and a row and column parameters and returns a 2D row by col array with values filled in by the 1D array. If there are not  enough values in the 1D array, then remaining entries in the 2D array = 0. If too many entries in the 1D array, then only the entries that can fit in the 2D array will be placed. |
| 14 | MergeSortedArray | public String[] createMerge() | Merge sorted arrays take two arrays each which is sorted ascending and returns an array that is also sorted ascending and contains the elements of both arrays. Sorting is case insensitive |
| 15 | StringSelectionSortIgnoreCase | public void stringSelectionSortIgnoreCase() | String Selection Sort Ignore Case takes an array of Strings and uses selection sort to create a sorted array. The sort should be case insensitive. You may not use Arrays.sort() |
| 16 | StringShuffle | public void shuffleList() | String Shuffler takes an ArrayList of Strings and shuffles them randomly like a deck of cards should be. Test data (of course your results will differ) |
| 17 | TallyLetters | public int[] tallyLetterArray() | Tally letters takes a String as input and returns an array that tallies the number of times each letter appears in the String. Each appearance of 'a' or 'A' is tallied at index 0, 'b' or 'B" is tallied in index 1, ... and 'z' or 'Z' in index 25. Non-letters are not counted. |
| 18 | UniqueStrings | public ArrayList<String> listUniqueStrings() | Unique Strings takes an array of Strings as input and returns an ArrayList containing the unique strings (case-sensitive) in the array. The resulting ArrayList should be in alphabetical (case-sensitive) order. You may not use the ArrayList contains() method in  your solution. |
| 19 | Generate12DaySong | public String generateSong() | Generate 12 days song creates a parody song similar to the "12 days of Christmas". Inputs are a festival (e.g. "Christmas"), benefactor (e.g. "true love"), and an array of gifts. |
| 20 | BooleanAdder | private boolean[] addBooleanArrays() | Boolean Adder takes in two strings consisting of ones and zeros (binary numbers) and  returns a string that returns a string consisting of ones and zeros that represent the addition of the two binary numbers. Internally, these Strings of 0 and 1s are converted to and array of  booleans for input/output. You may not use the Integer.parseInt() to solve this problem. |
| 21 | CountWords | public int countWords() | File Word Count takes a file name as input and returns the number of words/tokens in the file. Note that tokens are divided by one or more "white space" characters and that a file with no path information is in the root of the project or in the same folder as the jar file. |
| 22 | BinaryIntegerSearch | public int binarySearch() | Binary Search Int takes a target number and an increasing array of int  It returns the index of an occurrence of target if the target is in the increasing array.  Otherwise, it returns the negative of one more than the last index tested. You may not use the Arrays.binarySearch() method to solve this problem. |
| 23 | GreatestCommonDivisor | public int calculateGCD() | Greatest Common Divisor takes two numbers and finds the greatest common divisor of the  two numbers. While other algorithms may be used, suggest using [Euclid's algorithm](https://en.wikipedia.org/wiki/Euclidean_algorithm.). |
| 24 | HalfCollision | public int calculateHalfCollision() | Given a uniform distribution of slots/hashes/values for "hits", how many "hits" can you have before there is at less than a 50 percent probability that every hit is on a different slot.  This probability = 1- (hashes - 1)/hashes \* (hashes -2)/hashes \* ... (hashes - hits)/hashes. |
| 25 | MinesweeperCalc | public int[][] calculateMineCounts() | Minesweeper Calc takes a two dimensional array of 1s and 0s where the 1s are mines and the 0s are safe and returns a two dimensional array of the  same dimensions where each mine is represented by -1 and all other blocks count the number of mines (up to 8) around it. |
| 99 | SampleProblem | public String reverseString() | This is the sample problem. This particular class/method takes a String as input  and returns the String with all the characters reversed. |

# Sample output for “Stacking Blocks”

Parameter = 3

+-+

|1|

+-+

+--+

|2 |

| 2|

+--+

+---+

|3 |

| 3 |

| 3|

+---+