Efi Arazi School of Computer Science

Introduction to Computer Science

Midterm Examination 2021

* The exam lasts 2 hours. There will be no time extension.
* Use your time efficiently. If you get stuck somewhere, leave the question and move on to another question.
* Use of digital devices, books, lecture notes, and anything other than the exam pages is forbidden. All the materials that you need for answering this exam are supplied with the exam. You can use a paper-based dictionary, if you want.
* Answer all questions on exam pages.
* **Answer all the questions on the exam pages, using a pen. Don’t write anything on the back of the pages.** Only the front pages are scanned for grading. You can use blank pages for draft (טיוטה), which will not be graded.
* You can answer any question in either English or Hebrew.
* If you feel a need to make an assumption, you may do so as long as the assumption is reasonable and clearly stated.
* If you can't give a complete answer, give a partial answer. A partial answer will award partial points.
* If you are asked to write code and you feel that you can't write it, you may describe what you wish to do in natural language (English or Hebrew). A good explanation will award partial credit.
* If you are asked to write code that operates on some input, there is no need to validate the input unless you are explicitly asked to do so. Likewise, if you are asked to write a function that operates on some arguments, there is no need to validate the arguments unless you are explicitly asked to do so.
* There is no need to document the code that you write, unless you want to communicate something to us.
* The code that you write will be judged, among other things, on its conciseness, elegance, and efficiency. Unnecessarily long or cumbersome code will cause loss of points, even if it provides the correct answer.
* No points will be taken for trivial syntax errors. For example, instead of writing System.out.println(x) you can write println(x).

Good Luck!

Consider the following program:

public class Mystery {

public static void main(String[] args) {

int x = 3;

int[] y = { 1, 2, 3 };

triple1(x);

triple2(y);

System.out.println(x);

for (int i = 0; i < x; i++) {

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

}

}

public static void triple1(int a) {

a = 3 \* a;

}

public static void triple2(int[] a) {

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

a[i] = 3 \* a[i];

}

}

}

1. (10 points)

(a) Write the program’s output below. (b) Explain your answer.

Output:

Explanation:

Questions 2, 3, 4, 5 deal with the class Sets, described in the help page.

Take five minutes to read this help page now.

2. (10 points) Implement the following function. Your code can use any one of the Sets class functions listed in the help page, even if the function is not implemented.

/\*\* Returns the intersection of the three sets. The intersection of set1, set2, and set3

\* is the set containing the elements that are both in set1, set2, and set3. \*/

public static int[] intersection (int[] set1, int[] set2, int[] set3) {

// Write your code here:

3. (5 points) The Sets class includes two functions that have the same name (intersection).

(a) How does the Java compiler know which function to call?

(b) What is the name of this programming technique?

(c) Is it recommended to use this programming technique? Why or why not?

4. (15 points) Implement the following function. Your code can use any one of the Sets class functions listed in the help page, even if the function is not implemented. Tip: This problem can be solved efficiently by going once over all the elements of one set, and doing something, and once over all the elements of the other set, and doing something. Try to find and implement such an algorithm.

/\*\* Checks if set1 dominates set2. Set1 dominates set2 if every element of set1

\* is strictly greater (>) than every element of set2. \*/

public static boolean dominates(int[] set1, int[] set2) {

// Write your code here:

5. (20 points) Implement the following function.

/\*\* Returns a set containing the odd elements of the given set (elements that are not multiples of 2). \*/

public static int[] oddValuesOf(int[] set) {

// Write your code here:

Question 6 deals with the class Tables, described in the help page.

Take five minutes to read this help page now.

6. (20 points) Implement the following function.

/\*\* Returns a one-dimensional array containing the averages of the columns of the given

\* table (a two-dimensional array in which all rows have the same number of elements). \*/

public static double[] colAverages(int[][] arr) {

// Write your code here:

7. (20 points) In HW4 you had to implement a function that takes a String consisting of digit characters, like "536", and returns the int value 536. The intToString function performs the opposite operation: it takes an int value, like 536, and returns the string "536". For example, the expression intToString(73) + intToString(73) results in the string "7373".

Implement the function. If you think that you need it, you may consult the ASCII table given in the help pages. Note: A trivial solution is to simply “add” the given number to an empty string. You have to implement the real thing, which is handling each digit separately and building the string incrementally, one step at a time.

/\*\* Returns the string representation of the given integer value. Assumes that the integer is greater than 0. \*/

public static String intToString(int x) {

// Write your code here: