**CSC 3020 – Java Programming**

**Homework 3 – [Sayem Chowdhury]**

**25 points – Due February 21, 10am**

**Late deadline is February 23, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document and your .java file(s) to the Blackboard item where you downloaded this document. Do not submit a zip file but individually attach your files.

**1) [10 points]** You've been hired by *Sort Scorchers* to write a Java console application that sorts an **array list** of random real numbers. Use a validation loop to prompt for and get from the user the number of real numbers to sort in the range 10-10,000. Create and store in the array list that many random real numbers. Randomly generate the real numbers in the range 0-1,000. Print the array list size and use formatted output to print the first ten values of the array list in two columns:

● The first column is the zero-based index of the value.

● The second column is the right-justified value.

Use the insertion sort method to sort the array list. Monitor the number of cycles and swaps during the sort, and the elapsed time in milliseconds to complete the sort. Use formatted output to print the results (cycles, swaps, total cycles and swaps, and elapsed time) in two columns:

● The first column is a left-justified label. Include units if needed.

● The second column is a right-justified value.

Print the array list size and first ten values of the array list again. Format all numbers with a comma. Format all real numbers to one decimal place. Run the program three times with the following inputs. Enter the elapsed times for each run:

|  |  |  |
| --- | --- | --- |
| Run | Array list size | Elapsed time (ms) |
| 1 | 10 |  |
| 2 | 5,000 |  |
| 3 | 10,000 |  |

*[your program code here*

*//======================================================================*

*// Title: <Array\_List\_rand.java> // Sort an Array list of random numbers*

*// Course: CSC 3020*

*// Homework: <#3)>*

*// Author: <Sayem Chowdhury>*

*// Date: <2/22/2018>*

*// Description:*

*/\*<*

*A Java console application that sorts an array list of random real numbers.*

*Using a validation loop to prompt for and get from the user the number of real numbers to sort in the range 10-10,000.*

*To create and store in the array list that many random real numbers. Randomly generate the real numbers in the range 0-1,000.*

*Also Print the array list size and use formatted output to print the first ten values of the array list in two columns:*

*The first column is the zero-based index of the value.*

*The second column is the right-justified value.*

*>\*/*

*//*

*//======================================================================*

*//package Name*

*package sort\_scorchers;*

*//import Classes*

*import java.util.Random;*

*import java.util.Scanner;*

*import java.util.ArrayList;*

*public class Array\_List\_rand {*

*// [static /global variables]*

*static int cycles=0;*

*static int swaps=0;*

*static int total;*

*// Array Size*

*public static int Random\_num=10;*

*// Declare Format constants*

*private static final String COLFMTD1 = "%2d";*

*private static final String COLFMTF = "%20.1f";*

*private static final String COLFMTS1 = "%-16s";*

*private static final String COLFMTS2 = "%6s";*

*//------------------------------------------------------------------*

*// randomizeArray*

*//------------------------------------------------------------------*

*private static void randomizeArray(*

*ArrayList<Double>arr, int upperLimit,int lowerLimit)*

*{*

*// Loop to generate and store random numbers in array*

*Random rand = new Random();*

*for (int i = 0; i < Random\_num; i++)*

*{*

*arr.add((double)(lowerLimit+rand.nextInt(upperLimit-lowerLimit) + 1));*

*}*

*}*

*//print function will display sorted and unsorted Array*

*private static void printArray(String heading, ArrayList<Double>arr)*

*{*

*System.out.println("\n" + heading + " values.");*

*System.out.println("Array size: " + Random\_num);*

*System.out.println("-----------------------------------");*

*System.out.printf(COLFMTS1 + COLFMTS2 + "%n", "Index", "Value");*

*// Loop to print array numbers*

*for (int i = 0; i < 10; i++)*

*System.out.printf(COLFMTD1 + COLFMTF + "%n", i, arr.get(i));*

*}*

*// Insertion sort Method*

*public static void insertion\_sort( ArrayList<Double>arr )*

*{*

*//-----------------*

*int L = arr.size();*

*int i, j;*

*double temp;*

*for (i = 1; i< L; i++)*

*{*

*j = i;*

*temp = arr.get(i);*

*while (j > 0 && temp < arr.get(j-1))*

*{*

*arr.set(j,arr.get(j-1));*

*j = j-1;*

*swaps=swaps+1; // counting swaps*

*cycles=cycles+1;//counting cycles*

*}*

*arr.set(j,temp);*

*}*

*total=swaps+cycles; // total numbers of swap and cycles*

*}*

*//-----------------------------------------------------------------------------------*

*//main function*

*public static void main(String[] args) {*

*// Input Variables*

*Scanner keyboard = new Scanner(System.in);*

*// Show application header*

*System.out.println("Welcome to Sorting Array List Application");*

*System.out.println("-----------------------------------------");*

*// Validation for Array Size*

*while(true)*

*{*

*System.out.print("Please Enter the Array Size between 10 to 10000 \n");*

*Random\_num=keyboard.nextInt();*

*if((Random\_num<10 || Random\_num>10000))*

*{*

*System.out.print("Please Enter the Array Size between 10 to 10000");*

*Random\_num=keyboard.nextInt();*

*}*

*else*

*{*

*break;*

*}*

*}*

*//Declaring Array List*

*ArrayList<Double>unsortedArray = new ArrayList<>();*

*ArrayList<Double>sortedArray = new ArrayList<>();*

*ArrayList<Double>insertionSortedArray =new ArrayList<>();*

*//variable for calculating times take by insertion method to sort the Array LIst*

*long before;*

*long after;*

*// Declare variables*

*int lowerBound;*

*int upperBound;*

*// Prompt for user input of the upper and lower bound and get random range*

*System.out.print("\nEnter the lower bound: ");*

*lowerBound = keyboard.nextInt();*

*System.out.print("Enter the upper bound: ");*

*upperBound = keyboard.nextInt();*

*//Validation Loop*

*if(lowerBound<10 || upperBound>1000)*

*{*

*System.out.print("Out of range");*

*System.exit(1);*

*}*

*// Place random numbers in array and print*

*randomizeArray(unsortedArray, upperBound,lowerBound);*

*//printing Unsorted Array using print function*

*printArray("Unsorted Array", unsortedArray);*

*//copying*

*for (int i = 0 ; i<unsortedArray.size();i++){*

*sortedArray.add(unsortedArray.get(i)) ;*

*}*

*//sorting sortedArray array*

*insertion\_sort(sortedArray );*

*//printing sorted array using print function*

*printArray("Sorted Array", sortedArray);*

*//copying*

*for (int i = 0 ; i<unsortedArray.size();i++){*

*insertionSortedArray.add(unsortedArray.get(i)) ;*

*}*

*//staring time of insertion sort method*

*before = System.currentTimeMillis();*

*insertion\_sort(insertionSortedArray);*

*//Ending time of insertion method*

*after = System.currentTimeMillis();*

*//------*

*// Print headings*

*System.out.println();*

*System.out.println("------------------------------------------------------------------------------------");*

*System.out.printf("%n%-10s %15s %n%-10s%,17d %n%-10s %,16d %n%-10s%,17d %n%-10s%17d %n", "Sort method","Insertion",*

*"Swaps", swaps,"Cycles",cycles,"Total",total,"Time (ms)",(after-before));*

*System.out.println("------------------------------------------------------------------------------------");*

*// close Keyboard*

*keyboard.close();*

*}*

*}*

*]\**

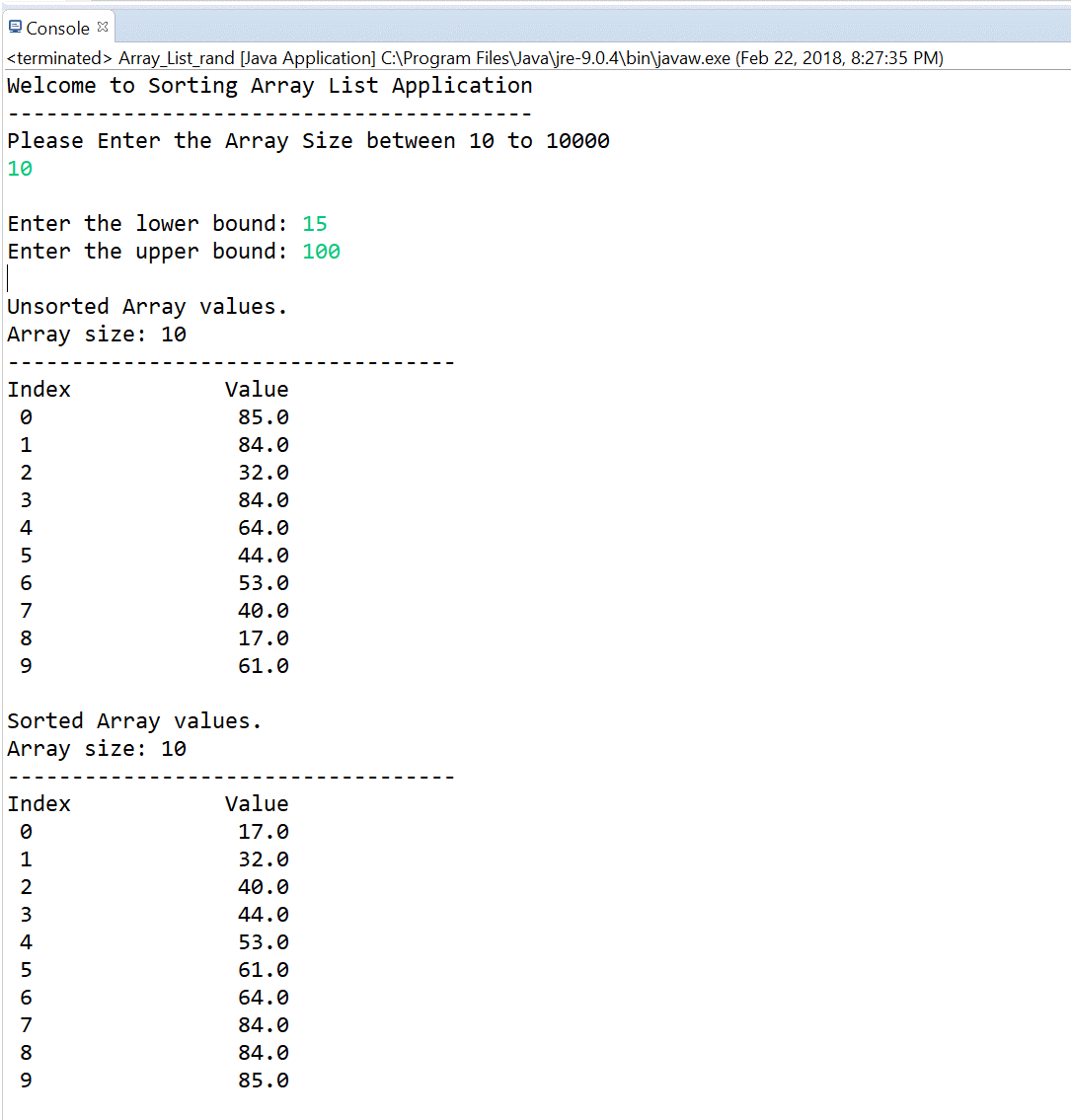
**If possible, format your code like this:**

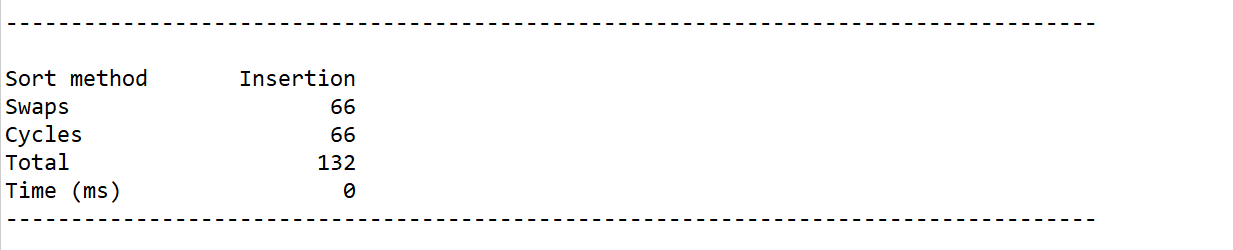
**Font “Courier New”**

**Font size “9”**

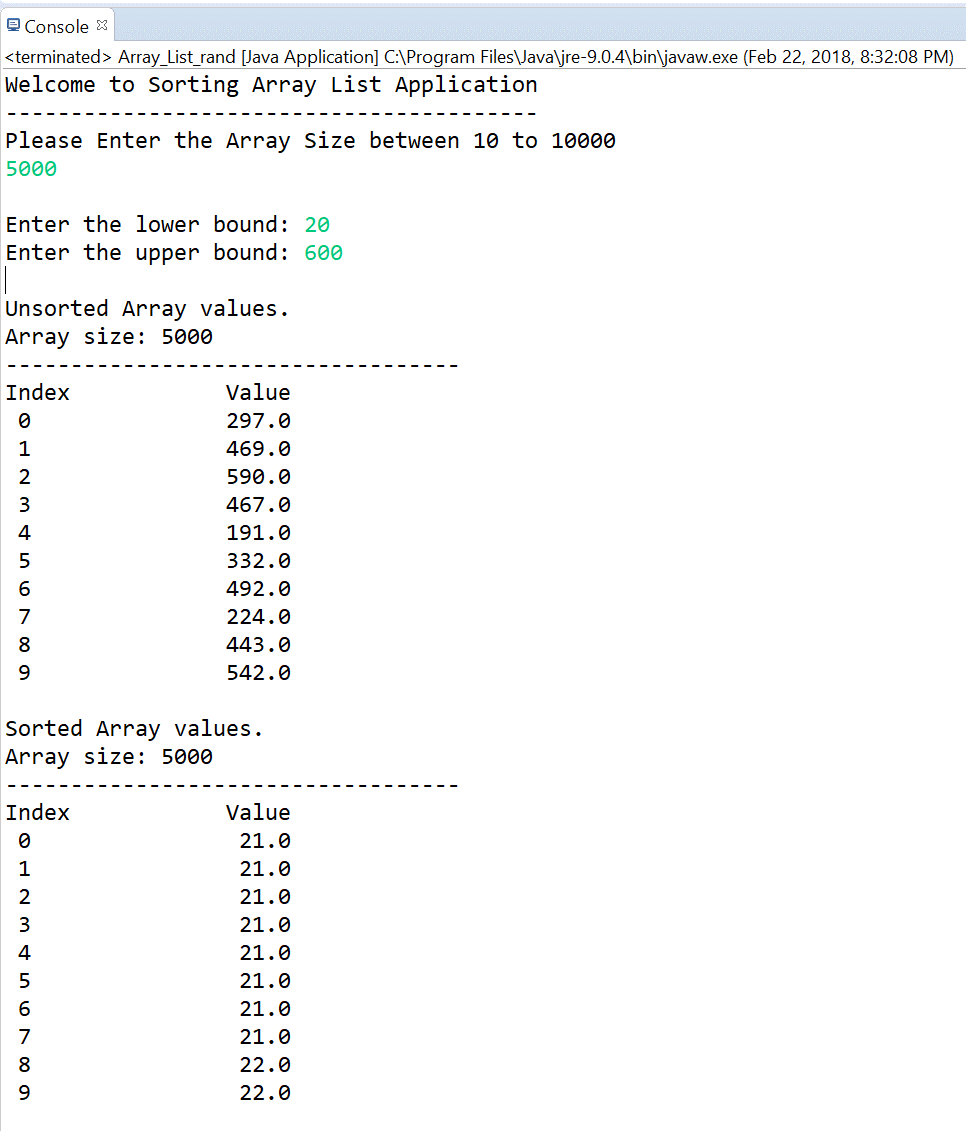
**Bold**

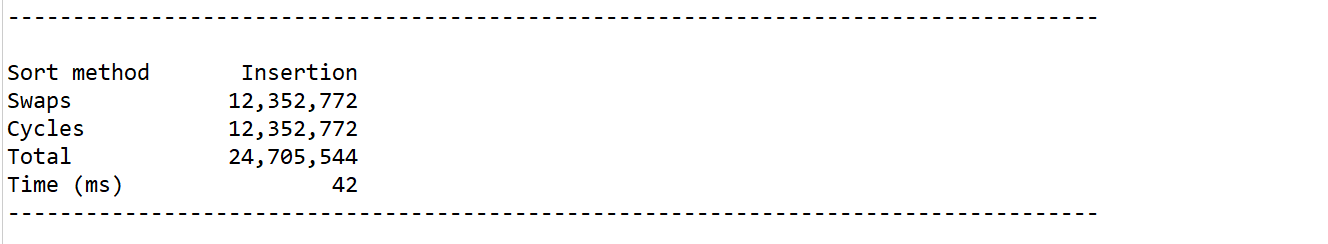
*[your program output here – one screenshot per run*



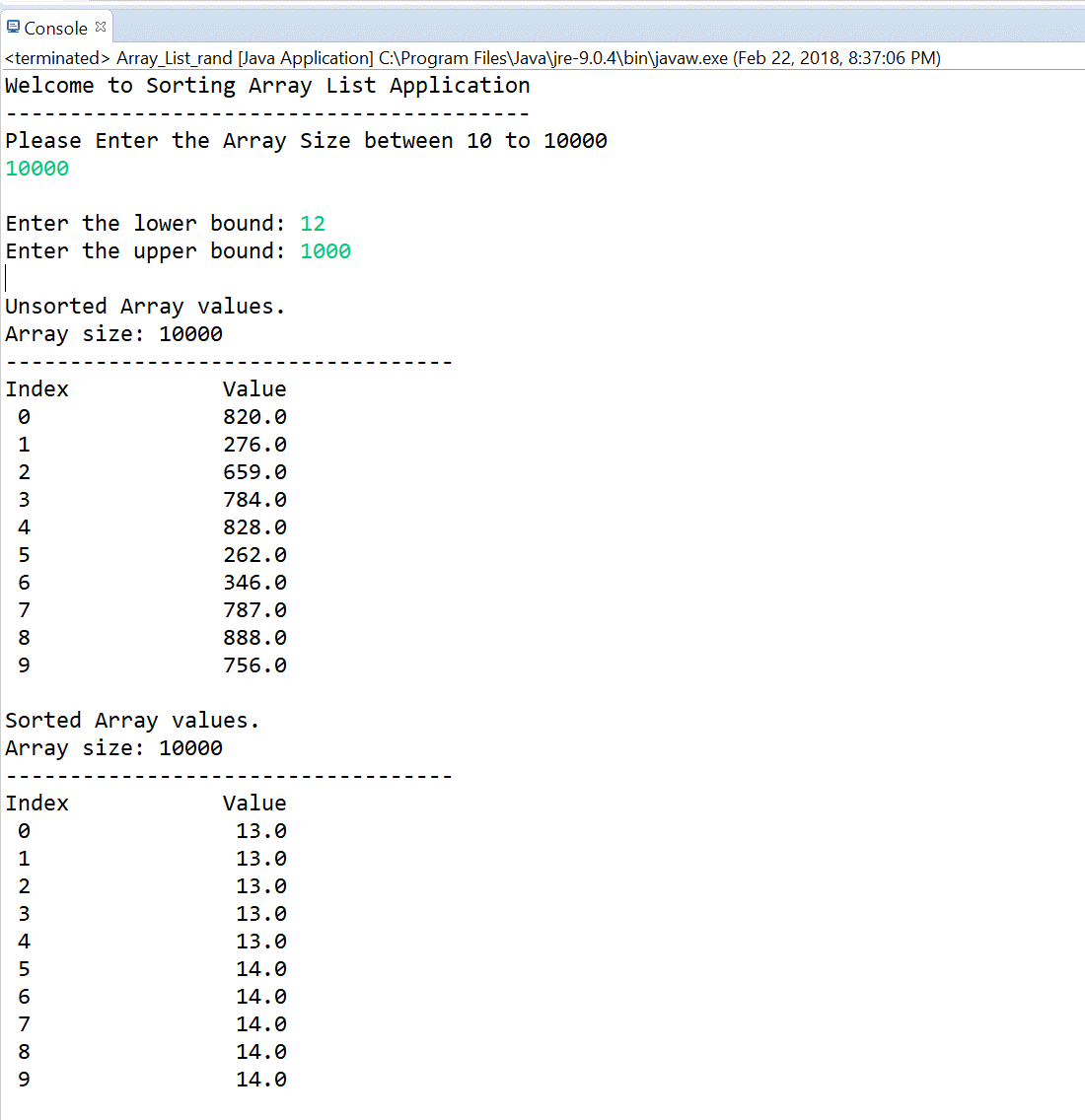


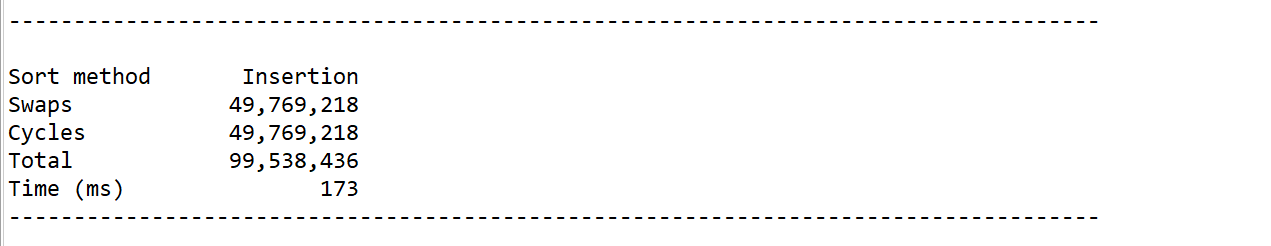
*//*





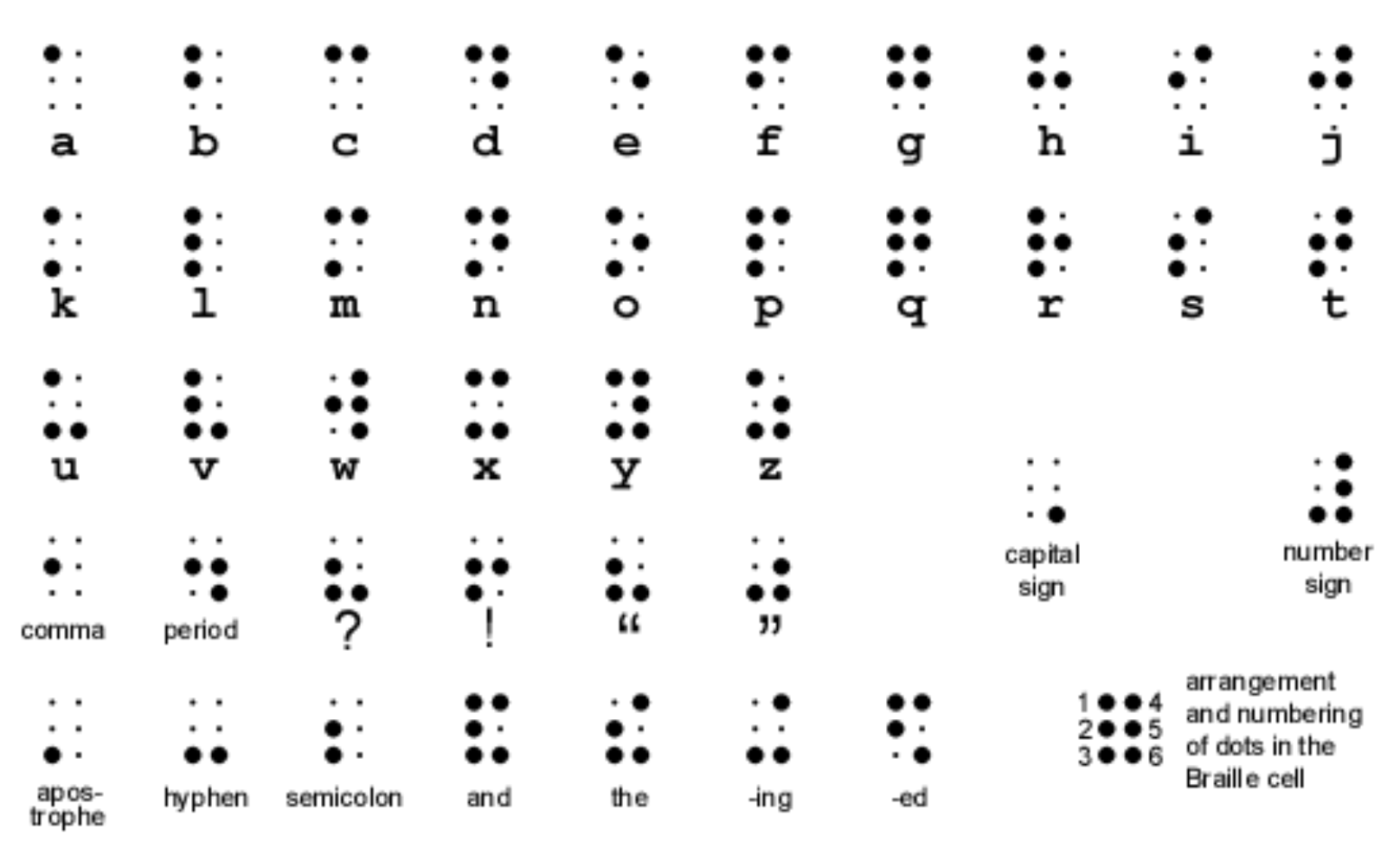
*//-------------------------------------------*





*]\*\**

**2) [15 points]** You've been hired by *Braille Buddies* to write a Java console application that prints the equivalent Braille of the text entered by the user. Braille is a special language used by the blind to read text. It uses a 3x2 grid of dots to represent different characters. Here are the grid definitions for Grade 1 Braille:



To specify an upper case letter, the "capital sign" grid defined above precedes a lower case letter. Use the following three-dimensional array declaration to represent the Braille grid definitions. The first dimension is the letter, the second dimension is the row, and the third dimension is the column of a spot in a grid. A one (1) indicates a big spot while a zero (0) indicates a small spot:

final int[][][] BRAILLE\_GRIDS =

{

{{ 1, 0}, { 0, 0}, { 0, 0}}, // 'a' - 0

{{ 1, 0}, { 1, 0}, { 0, 0}}, // 'b' - 1

{{ 1, 1}, { 0, 0}, { 0, 0}}, // 'c' - 2

{{ 1, 1}, { 0, 1}, { 0, 0}}, // 'd' - 3

{{ 1, 0}, { 0, 1}, { 0, 0}}, // 'e' - 4

{{ 1, 1}, { 1, 0}, { 0, 0}}, // 'f' - 5

{{ 1, 1}, { 1, 1}, { 0, 0}}, // 'g' - 6

{{ 1, 0}, { 1, 1}, { 0, 0}}, // 'h' - 7

{{ 0, 1}, { 1, 0}, { 0, 0}}, // 'i' - 8

{{ 0, 1}, { 1, 1}, { 0, 0}}, // 'j' - 9

{{ 1, 0}, { 0, 0}, { 1, 0}}, // 'k' - 10

{{ 1, 0}, { 1, 0}, { 1, 0}}, // 'l' - 11

{{ 1, 1}, { 0, 0}, { 1, 0}}, // 'm' - 12

{{ 1, 1}, { 0, 1}, { 1, 0}}, // 'n' - 13

{{ 1, 0}, { 0, 1}, { 1, 0}}, // 'o' - 14

{{ 1, 1}, { 1, 0}, { 1, 0}}, // 'p' - 15

{{ 1, 1}, { 1, 1}, { 1, 0}}, // 'q' - 16

{{ 1, 0}, { 1, 1}, { 1, 0}}, // 'r' - 17

{{ 0, 1}, { 1, 0}, { 1, 0}}, // 's' - 18

{{ 0, 1}, { 1, 1}, { 1, 0}}, // 't' - 19

{{ 1, 0}, { 0, 0}, { 1, 1}}, // 'u' - 20

{{ 1, 0}, { 1, 0}, { 1, 1}}, // 'v' - 21

{{ 0, 1}, { 1, 1}, { 0, 1}}, // 'w' - 22

{{ 1, 1}, { 0, 0}, { 1, 1}}, // 'x' - 23

{{ 1, 1}, { 0, 1}, { 1, 1}}, // 'y' - 24

{{ 1, 0}, { 0, 1}, { 1, 1}}, // 'z' - 25

{{ 0, 0}, { 0, 0}, { 0, 1}} // upper case follows - 26

};

Use a validation loop to prompt for and get from the user a string that contains only upper and lower case letters and spaces. If an invalid string is entered, print an error message. If a valid string is entered, convert and display it in Braille. Print the following four lines for each string:

|  |  |
| --- | --- |
| Line | Output |
| Character | ● For an lower case letter, print the letter.  ● For an upper case letter, print "UP" and the corresponding lower case letter.  ● For the space character, print a space. |
| Rows 1-3 | ● For lower case letters, print its grid.  ● For upper case letters, prints two grids: the "capital sign" grid and the corresponding lower case letter grid.  ● For the space character, print spaces. |

To access the correct grid in array BRAILLE\_GRIDS, convert a lower case letter to value between 0 and 25 (the first dimension) using the fact that each character is assigned a unique integer. Here is a run with a sample string:

Enter a string (letters and spaces only) to convert to Braille (q to exit): Hello World

Character: UP h e l l o UP w o r l d

Row 1: 0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 1

Row 2: 0 0 1 1 0 1 1 0 1 0 0 1 0 0 1 1 0 1 1 1 1 0 0 1

Row 3: 0 1 0 0 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 0

Enter a string (letters and spaces only) to convert to Braille (q to exit):

Continue to prompt the user for strings until they enter “q”. Use these strings for the last three inputs:

|  |  |
| --- | --- |
| Run | Text |
| 1 | The quick Fox |
| 2 | Jumps over |
| 3 | the Lazy Dog |

*[your program code here*

*/======================================================================*

*//*

*// Title: <B\_Readers> // BRAILLE READERS*

*// Course: CSC 3020*

*// Homework: <#3>*

*// Author: <Sayem Chowdhury>*

*// Date: <2/22/2018>*

*// Description:*

*/\*<*

*A Java console application that prints the equivalent Braille of the text entered by the user.*

*Braille is a special language used by the blind to read text. It uses a 3x2 grid of dots to represent different characters.*

*>\*/*

*//*

*//======================================================================*

*//Package Name*

*package b\_readers1234;*

*//Import Classes*

*import java.util.Scanner;*

*// Class B\_Readers*

*public class B\_Readers {*

*//main Function*

*public static void main(String[] args) {*

*// Show application header*

*System.out.println("Welcome to BRAILLE BUDDIES!");*

*System.out.println("--------------------------------\n");*

*// 3 Dimensional Array //Matrix*

*final int[][][] BRAILLE\_GRIDS =*

*{*

*{{ 1, 0}, // Inside of Each Level (3 \*2 ) dimensional Matrix*

*{ 0, 0},*

*{ 0, 0}}, // 'a' - 0*

*{{ 1, 0},*

*{ 1, 0},*

*{ 0, 0}}, // 'b' - 1*

*{{ 1, 1}, { 0, 0}, { 0, 0}}, // 'c' - 2*

*{{ 1, 1}, { 0, 1}, { 0, 0}}, // 'd' - 3*

*{{ 1, 0}, { 0, 1}, { 0, 0}}, // 'e' - 4*

*{{ 1, 1}, { 1, 0}, { 0, 0}}, // 'f' - 5*

*{{ 1, 1}, { 1, 1}, { 0, 0}}, // 'g' - 6*

*{{ 1, 0}, { 1, 1}, { 0, 0}}, // 'h' - 7*

*{{ 0, 1}, { 1, 0}, { 0, 0}}, // 'i' - 8*

*{{ 0, 1}, { 1, 1}, { 0, 0}}, // 'j' - 9*

*{{ 1, 0}, { 0, 0}, { 1, 0}}, // 'k' - 10*

*{{ 1, 0}, { 1, 0}, { 1, 0}}, // 'l' - 11*

*{{ 1, 1}, { 0, 0}, { 1, 0}}, // 'm' - 12*

*{{ 1, 1}, { 0, 1}, { 1, 0}}, // 'n' - 13*

*{{ 1, 0}, { 0, 1}, { 1, 0}}, // 'o' - 14*

*{{ 1, 1}, { 1, 0}, { 1, 0}}, // 'p' - 15*

*{{ 1, 1}, { 1, 1}, { 1, 0}}, // 'q' - 16*

*{{ 1, 0}, { 1, 1}, { 1, 0}}, // 'r' - 17*

*{{ 0, 1}, { 1, 0}, { 1, 0}}, // 's' - 18*

*{{ 0, 1}, { 1, 1}, { 1, 0}}, // 't' - 19*

*{{ 1, 0}, { 0, 0}, { 1, 1}}, // 'u' - 20*

*{{ 1, 0}, { 1, 0}, { 1, 1}}, // 'v' - 21*

*{{ 0, 1}, { 1, 1}, { 0, 1}}, // 'w' - 22*

*{{ 1, 1}, { 0, 0}, { 1, 1}}, // 'x' - 23*

*{{ 1, 1}, { 0, 1}, { 1, 1}}, // 'y' - 24*

*{{ 1, 0}, { 0, 1}, { 1, 1}}, // 'z' - 25*

*{{ 0, 0}, { 0, 0}, { 0, 1}} // upper case follows - 26*

*};*

*// user Input Variables*

*Scanner KBD = new Scanner(System.in);*

*// Prompting for user input (a string )*

*System.out.print("Please Enter a and for exit please choose -1: ");*

*String ST = KBD.nextLine();*

*if(ST.equals("-1"))*

*System.exit(0);*

*//checking validation of sentence*

*String Sp=ST.replaceAll("\\s+", "");*

*//System.out.print(Sp);*

*boolean n=Sp.matches("[a-zA-Z]\*");*

*//System.out.print(n);*

*while(!n)*

*{*

*System.out.print("You Entered Invalid Sentence\n");*

*System.out.print("Please Enter a Sentence: ");*

*ST = KBD.nextLine();*

*if(ST.equals("-1") )*

*System.exit(0);*

*Sp=ST.replaceAll("\\s+", "");*

*boolean p=Sp.matches("[a-zA-Z]\*");*

*if (p)*

*n=true;*

*}*

*System.out.println("-----------------------------------------------------------------------------------------------\n");*

*String token;*

*//converting string to lower case string*

*token = ST.toLowerCase();*

*System.out.print("\nChracter: ");*

*for (int i = 0; i<ST.length();i++)*

*{*

*if ('A' <=ST.charAt(i) && ST.charAt(i)<= 'Z')*

*{*

*System.out.printf("%-10s","UP");*

*System.out.printf("%-5s",(token.charAt(i)));*

*}*

*else*

*System.out.printf("%-11s",ST.charAt(i));*

*}*

*//calling function print*

*print(BRAILLE\_GRIDS, ST);*

*// close KBD*

*KBD.close();*

*}*

*// Function Print Will display Braille Letters*

*private static void print(int [][][] ARRAY, String stg)*

*{*

*System.out.println("\nROW(1): \t");*

*for(int m=0; m<stg.length(); m++)*

*{*

*if ('A' <=stg.charAt(m) && stg.charAt(m)<= 'Z')*

*{*

*System.out.printf("%12s%s", ARRAY[26][0][0]," ");*

*System.out.print(ARRAY[26][0][1]);*

*char CH = Character.toLowerCase(stg.charAt(m));*

*int ASKI = ((int)CH - 97);*

*System.out.printf("%12s%s", ARRAY[ASKI][0][0]," ");*

*System.out.print(ARRAY[ASKI][0][1]);*

*}*

*else if(stg.charAt(m) == ' ') // checking for white space character in string*

*{*

*System.out.print(" ");*

*}*

*else*

*{*

*System.out.printf("%12s%s", ARRAY[(int)stg.charAt(m)-97][0][0]," ");*

*System.out.print(ARRAY[(int)stg.charAt(m)-97][0][1]);*

*}*

*}*

*System.out.println("\nROW(2): ");*

*for(int i=0; i<stg.length(); i++)*

*{*

*if ('A' <=stg.charAt(i) && stg.charAt(i)<= 'Z') // checking if there is upper case letter in string*

*{*

*System.out.printf("%12s%s", ARRAY[26][1][0]," ");*

*System.out.print(ARRAY[26][1][1]);*

*char ch = Character.toLowerCase(stg.charAt(i));*

*int AS = ((int)ch - 97);*

*System.out.printf("%12s%s", ARRAY[AS][1][0]," ");*

*System.out.print(ARRAY[AS][1][1]);*

*}*

*else if(stg.charAt(i) == ' ')*

*{*

*System.out.print(" ");*

*}*

*else*

*{*

*System.out.printf("%12s%s", ARRAY[(int)stg.charAt(i)-97][1][0]," ");*

*System.out.print(ARRAY[(int)stg.charAt(i)-97][1][1]);*

*}*

*}*

*System.out.println("\nROW(3): ");*

*for(int i=0; i<stg.length(); i++)*

*{*

*if ('A' <=stg.charAt(i) && stg.charAt(i)<= 'Z')*

*{*

*System.out.printf("%12s%s", ARRAY[26][2][0]," ");*

*System.out.print(ARRAY[26][2][1]);*

*char Ch = Character.toLowerCase(stg.charAt(i));*

*int CI = ((int)Ch - 97);*

*System.out.printf("%12s%s", ARRAY[CI][2][0]," ");*

*System.out.print(ARRAY[CI][2][1]);*

*}*

*else if(stg.charAt(i) == ' ')*

*{*

*System.out.print(" ");*

*}*

*else*

*{*

*System.out.printf("%12s%s", ARRAY[(int)stg.charAt(i)-97][2][0]," ");*

*System.out.print(ARRAY[(int)stg.charAt(i)-97][2][1]);*

*}*

*}*

*System.out.println("\n------------------------------------------------------------------------------------------------\n");*

*System.out.println();*

*System.out.print("\nEnd of the Application BRAILLE BUDDIES");*

*}*

*}*

*]\**

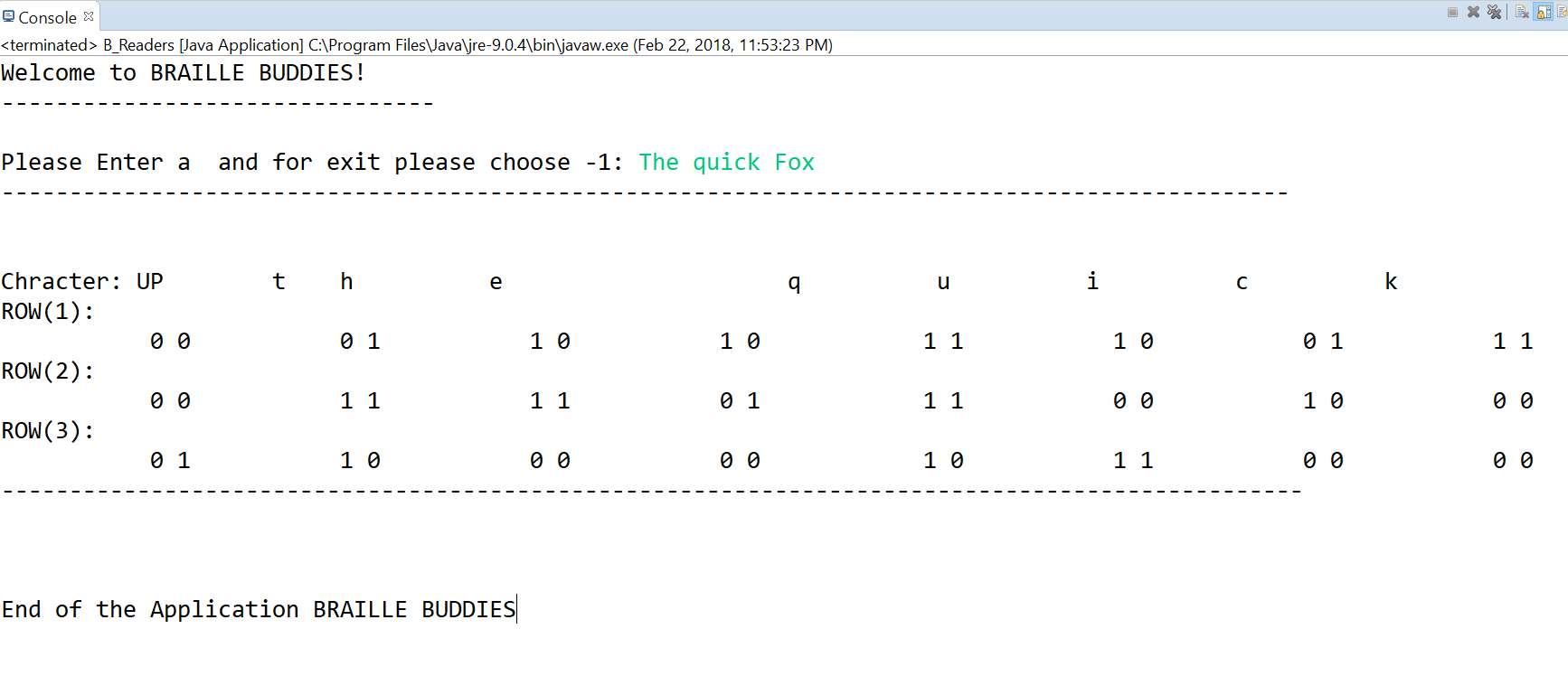
**If possible, format your code like this:**

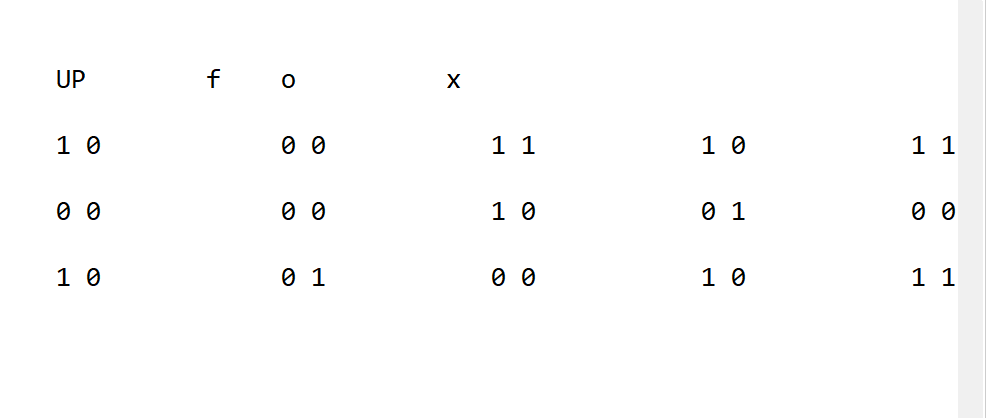
**Font “Courier New”**

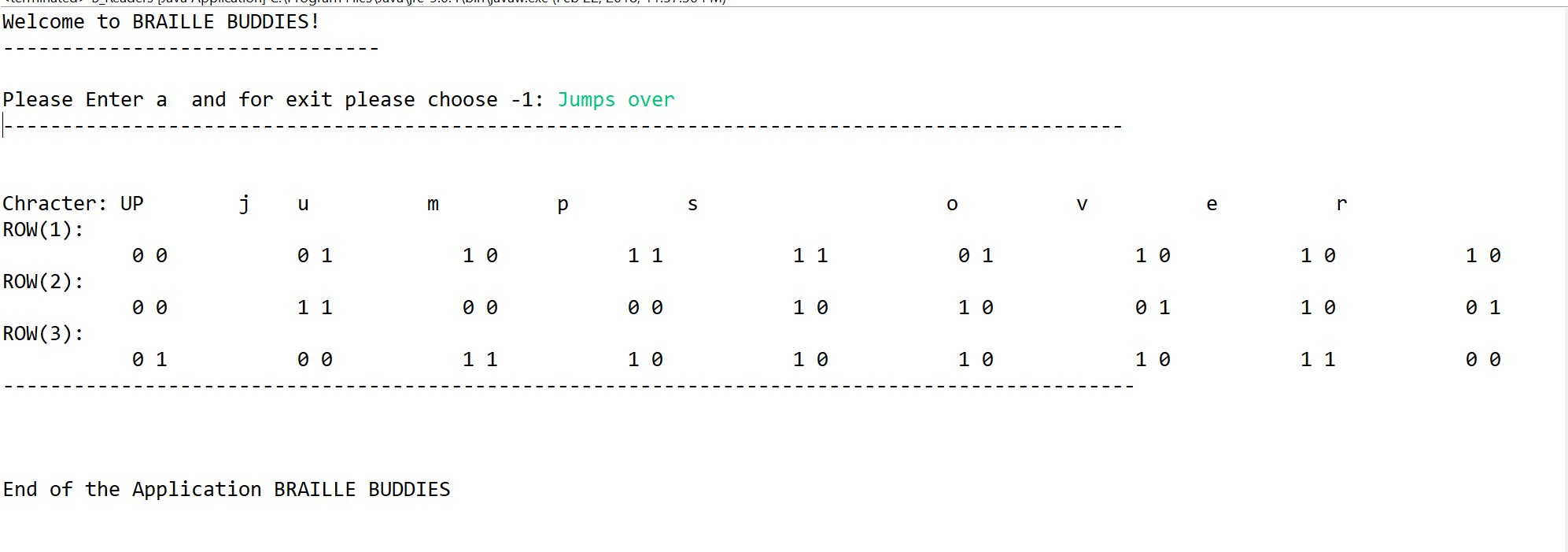
**Font size “9”**

**Bold**

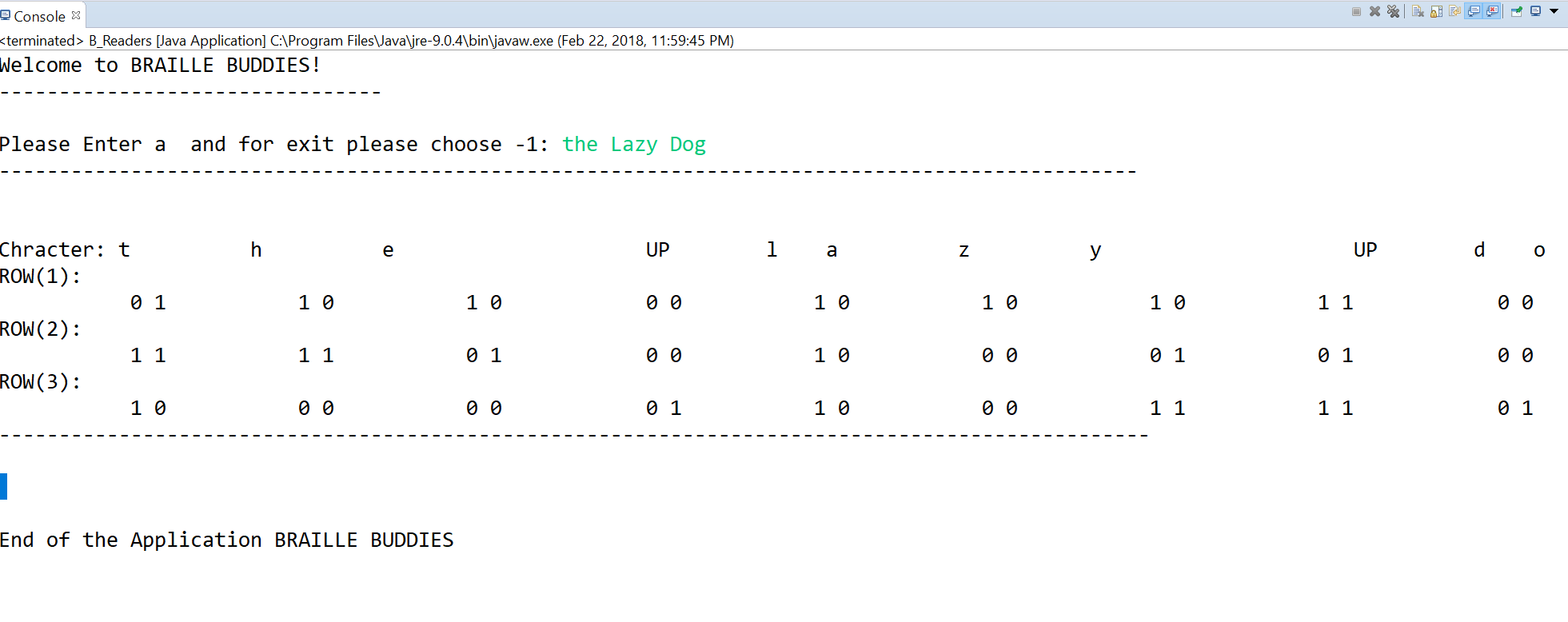
*[your program output here – one screenshot per run*

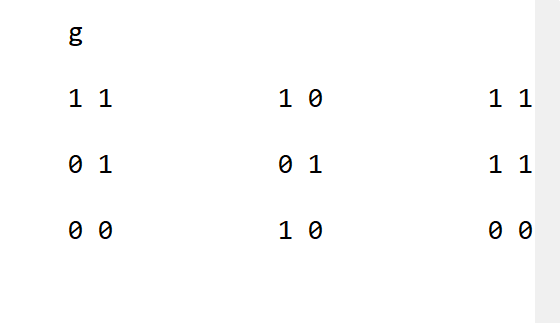






*//-----------------------------------------------------------------*



*///*

*]\*\**

\* **Copying-and-pasting application code to a Word document**

1) From the program editor window, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting application output to a Word document**

1) From the Eclipse main screen, maximize the Console window.

2) From the Console window, press **ALT-PrintScreen**.

3) From within the Word document, press **CTRL-V**.