

NAME: _____

1. (12) Write the Java code for a method named **maxChar** that receives two char arrays of the same size as arguments and returns a newly created array of type char. Each element of the new array that is returned is the larger of the corresponding elements of the two arrays received.

For example if ['x', '3', '?'] and ['7', '4', 'e'] were passed to **maxChar**, it would return the array: ['x', '4', 'e']. Or, if ['6', 'f', 'n'] and ['S', 'u', ','] were passed to **maxChar**, then ['S', 'u', 'n'] would be returned.

2. (12) Write a list of all the values that **s** is assigned in the course of the following computation:

```
String s = "Java Programming";  
int j = s.indexOf("av");  
int k = s.indexOf('m',j+1);  
s = s.substring(j,k);  
s = s+s.charAt(2)+s.charAt(s.length()-2);
```

NAME: _____

3. (10) Given the array of numbers below, show all the steps as you perform a binary search, looking for the **key** of **11**

```
int[] nums = {9, 15, 23, 67, 78, 85, 111, 117, 135, 225, 829, 1000};
```

Enter into the table to the right the low, mid and high values as the binary search progresses. DO NOT write any code. You may use either the subscript (index) values or the actual number stored at the subscript. Not all rows of the table below may be needed.

Low	Mid	High

4. (12)

The table to the right below shows the number of days in a month. Write, in Java, a method **numberOfDays** which accepts an **integer** (month) and **boolean** (isLeapYear) and returns an **integer** representing the number of days in the month. For example, if 3 and true are passed to **numberOfDays**, it will return 31; if 3 and false are passed, the method will return 31. If 2 and true are passed, it will return 29. If 2 and false are passed, it will return 28.

Month	Leap year	Maximum number of days
1	true or false	31
2	true	29
2	false	28
3	true or false	31
4	true or false	30
5	true or false	31
6	true or false	30
7	true or false	31
8	true or false	31
9	true or false	30
10	true or false	31
11	true or false	30
12	true or false	31

NAME: _____

5. (6) Convert the following mathematical expression into Java code (not a complete program).

$$W =$$

6. (6) Perform the following conversions.

a. 10101001 (base 2) to base 10 Answer _____

b. 1E (base 16) to base 10 Answer _____

c. 10001101 (base 2) to base 16 Answer _____

d. 77 (base 10) to base 16 Answer _____

e. 156 (base 10) to base 2 Answer _____

f. Perform the following addition of two binary (base 2) numbers.

Your answer should be a number in binary

101 + 001 = ? Answer _____

NAME: _____

7. (12) What does the following program print when executed?

```
public class Q7{
    public static void main(String[] args) {
        int[] v = {1,1,2,3,5,8};
        int[] w = new int[v.length];
        int n = 3, k=5;
        dump("HERE: v",v);
        System.out.println("n="+n+" k="+k);
        w=mangle(v,n);
        dump("THERE w",w);
        dump("EVERYWHERE: v",v);
        System.out.println("n="+n+" k="+k);
    }
    private static void dump(String label, int[] a) {
        System.out.print(label+": ");
        for (int i=0; i<a.length; i++){
            System.out.print(a[i]+" ");
        }
        System.out.println();
    }

    private static int[] mangle(int[] a, int k) {
        int[] b = new int[a.length];
        k++;
        for (int i=0; i<b.length-1;i++){
            b[i] = a[i+1];
        }
        b[b.length-1] = a[b.length-2] * 2;
        return b;
    }
}
```

NAME: _____

8. (30) Write a complete Java program. You must comment the **main** program and all methods, explaining what action(s) each performs.

Data: The input data for this program consists of the number of 2-point baskets and 3-point baskets each player scored in a game. At most there will be 25 players read in, though there could be less. All data will be read from a file (name of your choice) and all output will be written to a file (name of your choice).

Sample Data

8 3
6 5
4 2
9 1
12 0

A sample record consists of two numbers, the first representing the number of 2-point baskets and the second representing the number of 3-point baskets. A record 8 3 represents 8 2-point baskets (16 points) and 3 3-point baskets (9 points) (for a total of $8 \times 2 + 3 \times 3$ or 25 points).

main will invoke methods, that you write, to perform the following tasks:

(a) **main** will pass a filename (name of your choice) and two integer arrays to a method **readData**. The method will instantiate a Scanner object, using the filename passed, to read the number of 2-point and 3-point baskets per player and store them in two separate arrays. The method will return the total number of players read in.

For example, given the above data, after **readData** is finished executing, the arrays will contain:

2-point baskets	8	6	4	9	12
-----------------	---	---	---	---	----

3-point baskets	3	5	2	1	0
-----------------	---	---	---	---	---

and it will return 5

(b) **main** will invoke a method **totalPoints** that accepts a reference to an object of a class that will enable writing to a file (for example, **PrintWriter** or **PrintStream** or whichever you prefer), two integer arrays (the number of 2-point and 3-point baskets) and the number of players read in. It will return an integer array that contains, for each player, the total number of points that player scored.

Using the sample data above, the total points array will contain:

total points	25	27	14	21	24
--------------	----	----	----	----	----

The method will then print to the output file the number of 2 point baskets, 3 point baskets and total points for each player. The headers and the values are right justified.

2 pt baskets	3 pt baskets	total pts
8	3	25
6	5	27
4	2	14
9	1	21
12	0	24

(c) **main** will invoke a method **averages** that accepts a reference to an object of a class for output (same as passed to **totalPoints**), 3 integer arrays (2-point baskets, 3-point baskets, total points) and the number of records read in. It will compute and print to the output file a header (right adjusted) and the overall average number of points for 2-point baskets, the overall average number of points for 3-point baskets and the overall average number of total points. All averages will print right justified with a precision of 2. The output from the sample data would be:

avg2pt	avg3pt	avgpts
15.60	6.60	22.20

(d) **main** will then invoke a method **sort**, that is passed the array of total points, and sorts it in descending order. You must write a sort routine, do not use **Arrays.sort**

(e) **main** will print to the screen the number of points of the two highest scores and the two lowest scores. The first line, after the heading, will print the highest and lowest score, the next line will print the second highest and second lowest score. Using our sample data, the output will be:

High scores	Low scores
27	14
25	21

NAME: _____

EXTRA WORK SHEET: