



**The 9<sup>th</sup> Annual Newport News Computer Challenge**

**Thursday, February 18, 2010**

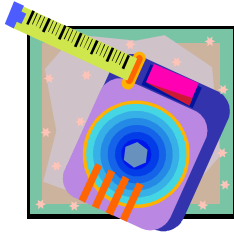
# Team Packet

# Java Problems



**The 9<sup>th</sup> Annual Newport News Computer Challenge**  
**Thursday, February 18, 2010**

## Java Problems



Distance 3D ~ 10 points



Credit Card Validation ~ 20 points



Freddie's Chair ~ 20 points

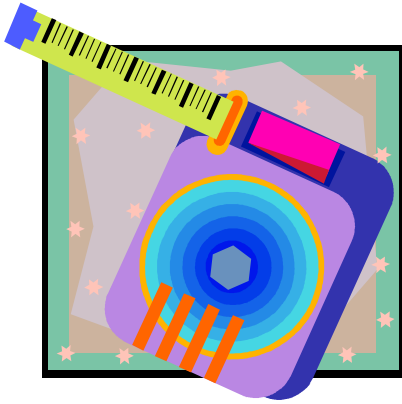


Sounds the Same ~ 30 points



## The 9<sup>th</sup> Annual Newport News Computer Challenge

Thursday, February 18, 2010



### Distance 3D (10 points)

Write a Java console program that calculates the distance between a pair of points on a three dimensional Cartesian coordinate system.

The distance  $d$  between two points,  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$ , on a three dimensional Cartesian coordinate system is

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

### Input Specification

Enter each of the six integer coordinates three per line. The first three integers represent the coordinates of one point on the coordinate system. The second three integers represent another point on the same system.

### Output Specification

After entering the coordinates, output a line that says “The distance from  $(x,y,z)$  to  $(a,b,c)$  is  $d.ddd$ ” where  $x,y,z,a,b,$  and  $c$  stand for the actual integers input by the user and  $d.ddd$  stands for the distance between those two points with three decimal places displayed (0’s if needed) and rounded to the nearest thousandth.

### Sample runs

```
Enter the three coordinates of the first point: 0 0 0
Enter the three coordinates of the second point: 0 0 0
The distance from (0,0,0) to (0,0,0) is 0.000
```

```
Enter the three coordinates of the first point: 1 -2 3
Enter the three coordinates of the second point: 1 -2 4
The distance from (1,-2,3) to (1,-2,4) is 1.000
```

*(more sample runs on next page...)*

Enter the three coordinates of the first point: 1 2 3  
Enter the three coordinates of the second point: 4 5 6  
The distance from (1,2,3) to (4,5,6) is 5.196

Enter the three coordinates of the first point: 18 22 72  
Enter the three coordinates of the second point: -122 -144 99  
The distance from (18,22,72) to (-122,-144,99) is 218.826



## The 9<sup>th</sup> Annual Newport News Computer Challenge

Thursday, February 18, 2010

### Credit Card Validation (20 points)

This problem lets you in on a little secret used by banks and lending institutions (called *the Luhn algorithm*, named after the IBM scientist Hans Peter Luhn who developed it in 1954) to



determine whether or not a credit card number is valid. What we're talking about here is not whether a credit card number is currently being used by someone, but rather whether or not the number **COULD** be used by someone, should a bank or lending institution issue that number.

Below is the Luhn algorithm that tests for credit card validity:

Step	Sample (using 5318-2795-1234-5678)
1. Reverse the order of the digits of the card number.	8765 4321 5972 8135
2. Beginning with the second digit and continuing with every other digit...	These digits are 7, 5, 3, 1, 9, 2, 1, and 5.
3. Perform this function on each of these digits: double the digit and (if the doubled result has two digits) add the digits of the doubled result.	$7 \times 2 = 14$ , $1 + 4 = 5$ $5 \times 2 = 10$ , $1 + 0 = 1$ $3 \times 2 = 6$ $1 \times 2 = 2$ $9 \times 2 = 18$ , $1 + 8 = 9$ $2 \times 2 = 4$ $1 \times 2 = 2$ $5 \times 2 = 10$ , $1 + 0 = 1$
4. Find the sum of these results.	The sum of 1, 2, 4, 9, 2, 6, 1, and 5 is 30.
5. Find the sum of the other digits.	The sum of 3, 8, 7, 5, 2, 4, 6, and 8 is 43.
6. Add the two sums.	30 plus 43 is 73.
7. If the final result is divisible by 10, the credit card number is valid. If not, it is invalid.	73 is not divisible by 10. Therefore the credit card number is invalid.

Your job, however, is not to test for validity, but rather to allow the user to enter the first ***n-1*** digits of a ***n***-digit credit card number and then output the digit that, if used as the ***n<sup>th</sup>*** digit, would make the card number valid. Input should allow dashes or spaces to be inserted anywhere in the number.

Samples:

First <i>n-1</i> digits	<i>n</i> th digit
5318-2795-1234-567	5
1234 12 123	2
000-000-100	8
000-000-1000	9



## The 9<sup>th</sup> Annual Newport News Computer Challenge

Thursday, February 18, 2010

### Freddie's Chair (20 points)

On certain days in Freddie's math class when too many students were absent due to SOL testing, Mrs. Magillicuddy would have the students who were left in class play a game she called "Chair Elimination".

The game of "Chair Elimination" works like this: The center of the room is cleared and students arrange their chairs in a circle in the center of the room, one for each student participating in the game, all chairs facing inward with nobody sitting in the chairs. Chairs are numbered in a clockwise direction starting with one and ending with the number of chairs.

The game leader (in this case, Mrs. Magillicuddy) randomly picks two numbers. The first number, called the "starter number", is a number of one of the chairs. The second number, called the "elimination number", is a number between 3 and 100 inclusive.

The game leader announces the two numbers and students are then told, "Ladies and gentlemen, have a seat," at which point everyone scrambles for a chair. The person who ends up in the chair whose number is the "starter number" is called the "starter".

Beginning with the "starter", students count aloud and clockwise around the circle. The "starter" says "one", the person to the immediate left of the "starter" says "two", the person to the immediate left of the person who said "two" says "three", and so forth. When some unfortunate student says the "elimination number", that student must get up, pick up his or her chair, and leave the circle. Chairs are scooted inward slightly to tighten up the circle again. The person who was on the immediate left of the person who got eliminated becomes the new "starter" and the counting process begins with "one" again. When all persons except one have been eliminated, the last remaining person becomes the winner.

Mrs. Magillicuddy always had plenty of great prizes for winners in this game such as pencils, pens, candy bars, and sometimes even a shiny new Susan B. Anthony silver dollar. But one day everyone realized that Freddie was winning almost all the time so Mrs. Magillicuddy said they couldn't play "Chair Elimination" anymore and that they had to do worksheets instead. What nobody realized was that Freddie had a programmable earring into which he could enter the number of chairs, the "starter number", and the "elimination number" and then calculate the number of the chair that would win. He would then quickly make his best effort to sit in that chair before anyone else.

Your job is to write the program that Freddie used to win. Please reject invalid input and only output the winning chair.

#### Samples:

10 chairs, starter number 4, elimination number 5

The eliminated chairs in order of elimination are 8 3 9 5 2 1 4 7 10 and the winning chair is 6.

15 chairs, starter number 9, elimination number 17

The eliminated chairs in order of elimination are 10 13 2 7 15 9 6 8 14 11 1 3 5 12 and the winning chair is 4.

You may receive up to 15 points for a program that runs correctly as specified and an additional 5 points (for a total of 20 points) if your program uses a GUI.



## The 9<sup>th</sup> Annual Newport News Computer Challenge

Thursday, February 18, 2010

### Sounds the Same (30 points)

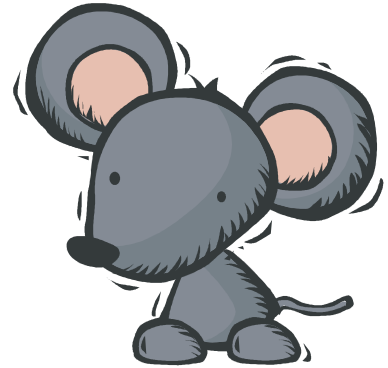
The soundex algorithm is used by spell checking programs to suggest words that sound like the word that the user typed. It works by encoding the original word. If a word in the spell check dictionary has the same code then it is a word to suggest to the user.

Your program will take as input two words in all capital letters. It will calculate the soundex encoding for each. Then it will tell whether the two words have the same soundex encoding. Finally it will tell the length of the common prefix.

#### The Algorithm

1. The first character in the encoding is the first letter of the word.
2. The following encoding is used on subsequent letters
  - a. discard all vowels and vowel-like letters: a,e,i,o,u,h,w,y
  - b. The remaining letters are mapped to digit characters:

b, p	=> 1
f, v	=> 2
c, k, s	=> 3
g, j	=> 4
q, x, z	=> 5
d, t	=> 6
l	=> 7
m, n	=> 8
r	=> 9
  - c. if two or more of the same digit are next to each other, discard all but the first.



#### Examples

```
input: APPLESAUCE APPLAUSE
output:
soundex of APPLESAUCE is A173
soundex of APPLAUSE is A173
they have the same encoding
they share a prefix of length 4
```

```
-----
input: BANANABOAT BANNER
output:
soundex of BANANABOAT is B816
soundex of BANNER is B89
they do not have the same encoding
they share a prefix of length 2
```

```
-----
input: MOTHER FATHER
output:
soundex of MOTHER is M69
soundex of FATHER is F69
they do not have the same encoding
they share a prefix of length 0
```



# The 9<sup>th</sup> Annual Newport News Computer Challenge

Thursday, February 18, 2010

## Java Ruberics for Teams

### Distance 3D (10 points)

Correct input/prompt: <i>Prompts and inputs 3 integers on one line and three on another.</i> <i>Sample:</i> Enter the three coordinates of the first point: 1 -2 3 Enter the three coordinates of the second point: 1 -2 4	1
Output line is correctly formatted: "The distance from (x,y,z) to (a,b,c) is d.ddd" <ul style="list-style-type: none"> <li>Attempts to echo input (1) <i>Values for all six points are there, but not correctly formatted.</i></li> <li>Correctly echoes input (1) <i>Values for all six points are there and correctly formatted up to and including the word "is":</i> "The distance from (x,y,z) to (a,b,c) is"</li> <li>Formats distance correctly (1) <i>Three decimal places rounded to the nearest thousandth, padded with zeros if needed.</i></li> </ul>	3
Correct distance	6
TOTAL	10

### Credit Card Validation (20 points)

Heading displays program name.	1
Friendly prompt to enter all but the last digit of a credit card number.	1
Accepts the input without crashing and without ending immediately.	2
Rejects or ignores invalid characters (only digits, spaces, and dashes are allowed).	3
For invalid input, displays a message stating that the input is invalid.	3
For valid input, correctly displays the last digit.	10
TOTAL	20

### Freddie's Chair (20 points)

Heading displays program name.	1
Prompts for number of chairs.	1
Rejects invalid number of chairs (1 or less).	1
Prompts for starter number.	1
Rejects invalid starter numbers (0 or less, greater than number of chairs).	1
Prompts for elimination number.	1
Rejects invalid elimination numbers (out of range of 3 to 100 inclusive).	1
Accepts all three valid numbers without crashing.	1
Displays winning chair <ul style="list-style-type: none"> <li>for all four samples below, 6 points</li> <li>for at least two samples below, 3 points</li> </ul>	6
Follows directions by only outputting the winning chair and not displaying the list of eliminated chairs. (No points unless at least two samples display the winning chair.)	1
Uses GUI	5
TOTAL	20

### Sounds The Same (30 points)

Heading displays program name and team's name.	1
Inputs two words per line without crashing.	4
Correctly displays soundex of first word.	5
Correctly displays soundex of second word.	5
Correctly displays whether or not word have the same encoding.	5
Correctly displays the shared prefix length.	5
Correctly displays all answers in format specified in sample runs.	5
TOTAL	30