

A+ Computer Science

October 2014

Computer Science Competition Hands-On Programming Set

DO NOT OPEN THIS PACKET UNTIL INSTRUCTED TO DO SO

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 15.
2. All problems have a value of 60 points. Incorrect submissions may be reworked and resubmitted, but will receive a deduction of 5 points for each incorrect submission. Deductions are only included in the team score for problems that are ultimately solved correctly.
3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.
4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

Number	Name
Problem 1	ReallyRandom
Problem 2	Nu
Problem 3	DailyDeal
Problem 4	FamilyFeud1
Problem 5	Composite
Problem 6	Initially
Problem 7	DNA1
Problem 8	Web
Problem 9	Trapped
Problem 10	AlsoPurchased
Problem 11	ConVow
Problem 12	Balance
Problem 13	Percent
Problem 14	DNA2
Problem 15	FamilyFeud2

Good luck!

1. ReallyRandom

Program Name: ReallyRandom.java

Input File: reallyrandom.dat

Before you travel to Vegas for a gambling trip, you practice your craps dice-rolling skills. You also test your dice to see if they are really random. There should be equal numbers of each roll. If you roll a die 100 times, you should get 16-17 rolls of each number. For this program, assume it is random if you get somewhere between 12% and 20% (inclusive) for each roll.

Given a list of numbers (1-6) determine if the dice rolls are random.

Input: The first line (N) consists of the number of data sets. Each data set contains 60 integers (1-6) with no spaces.

Output: Print out each either “YES” or “NO” for each data set.

Constraints:

$1 \leq N \leq 10$

Example Input File

3

```
111111111122222222223333333333444444444455555555556666666666
111111111111111111111111111111111111111111111111111111112
123456634321123456655321123456454321123456643321223456654322
```

Output to screen:

YES

NO

YES

2. Nu

Program Name: Nu.java

Input File: nu.dat

Sometimes it is hard to find humor in science class. That's why some of the jokes I tell do not go over too well. When we cover the speed of light, wavelength and frequency, I like to tell the following joke...

What does a nerd say when someone asks, "What's new?" "C over lambda!"

In the following equation, $c = \lambda \nu$,
 $c = 3.00 \times 10^8$ m/s, λ (lambda) is the wavelength in meters, and ν (nu) is the frequency in Hertz.

Given the wavelength, find the frequency. What's "nu"?

Input: The first line (N) consists of the number of data sets. Each data set contains 1 double type, the wavelength.

Output: Print out the frequency on one line in scientific notation (E notation). The answers should have 3 significant figures.

Constraints:

$1 \leq N \leq 10$

Example Input File

```
3
0.0000055
6.00e-7
1.20
```

Output to screen:

```
5.45E+13
5.00E+14
2.50E+08
```

3. DailyDeal

Program Name: DailyDeal.java

Input File: dailydeal.dat

I buy Kindle books all the time. However, I try to keep to a budget (about \$20/month) unless I get a Kindle gift card for birthday or Christmas—then I use it all up. I have a list of books and authors that I had in paperback which I’m replacing, so I look for specials. Amazon also has a “Kindle Daily Deal” where they offer several titles for \$1.99. It appears their marketing idea is to offer a nice price for one book. Then people will buy more of that author, or genre, or next book in a series. Amazon’s “One-Click Purchasing” is sometimes very tempting—keep them clicking! Write a program to see if this will profit Amazon.

You will be given a list of books purchased (one Daily Deal followed by 0 or more purchases). Amazon will have a profit if the purchased books price total at least \$20.00. Ignore tax for this program.

Input: The first line (N) consists of the number of data sets. Each data set starts with an integer T, the number of books bought, and then is followed by T book prices.

Output: Print out “YES” or “NO” to indicate if a profit is made by Amazon.

Constraints:

$1 \leq N \leq 10$

$0 < T \leq 10$

Example Input File

```
2
1
1.99
5
1.99
2.99
3.99
9.99
9.99
```

Output to screen:

```
NO
YES
```

4. FamilyFeud1

Program Name: FamilyFeud1.java

Input File: none

On the TV game show the Family Feud, when contestants give an answer not in the survey top answers, there is an X that shows up on the screen. You want to write a program that shows the X in a square shape, with a length of 5. The letter X will be a border on each side and along the diagonals, according to the output below.

Input: none

Output: Print out the picture below:

Output to screen:

```
XXXXXX
XX  XX
X  X  X
XX  XX
XXXXXX
```

5. Composite

Program Name: Composite.java

Input File: composite.dat

There is usually a program on prime numbers. So, this one is on composite numbers, numbers (greater than 3) that are not prime! Given a number, find the composite factors of it, showing all possible factors, or none if at least one of the pair of factors is prime. The first composite number is 4 followed by 6, 8, 9, 10...

40 has several factors: 1x40, 2x20, 4x10, 5x8. Only 4 and 10 are both composite.

8 has 1x8, 2x4. No factors are both composite.

60 has 1x60, 2x30, 3x10, 4x15, 5x12, 6x10. So 4x15 and 6x10 are both composite.

Input: The first line (N) consists of the number of data sets. Each data set contains one int, T.

Output: For each data set, print out the pairs of composite factors of T with “and” between them, or the word “none.” Separate multiple pairs with a comma, smallest factor first. For the sample and judge data, there will be no more than 7 pairs; so the output will fit on one line.

Constraints:

$1 \leq N \leq 10$

$4 \leq T < \text{Integer.MAX_VALUE}$

Example Input file

```
3
40
8
60
```

Output to screen:

```
4 and 10
none
4 and 15, 6 and 10
```

6. Initially

Program Name: Initially.java

Input File: initially.dat

I have almost all the John Grisham novels on my Kindle. I like to read them again and again because they are good books. The lawyers in his novels often have initials in their first name. For example, in The Street Lawyer, there is a character called N. Braden Chance: maybe his first name was Ned or Napoleon or something else he didn't like as a child. So in this program, change everyone's 3 word name to 1 initial + 2 word name like the following:

First Middle Last = F. Middle Last

Input: The first line (N) consists of the number of data sets. Each data set contains a name in the format First Middle Last.

Output: For each data set, print out the first initial (with period) and the middle and last names.

Constraints:

$1 \leq N \leq 10$

Example Input file

3

Ned Braden Chance
Josiah Willard Gibbs
Thomas Alva Edison

Output to screen:

N. Braden Chance
J. Willard Gibbs
T. Alva Edison

7. DNA1

Program Name: DNA1.java

Input File: dna1.dat

In DNA technology, palindromic DNA sequences are important. DNA palindromes are pairs of sequences in which the first strand going forward is the same as the other strand going backward (and have the correct A-T or C-G pairing). That is where restriction enzymes “cut” DNA strands. Molecular biologists can do many things with these strands.

Your task is to find palindromic sequences in a string of DNA base pairs. You will be given 2 lines of characters (ATCG) that are paired. For each double strand of DNA, find the position and length of palindromic sequences of at least 4 characters, if there is one at all.

Here is an example of palindromes:

```
AAAAAAAAAACCCCCCAATTCCCCCAAA
TTTTTTTTTTTGGGGGGGTTAAGGGGGTTT
123456789012345678901234567890
```

The AATT palindrome is the 18th-21st letters, so position 18 and 4 long. For this program, assume that the base pairs are correctly matched according to Chargaff’s Rules (A-T and C-G). There will at most one sequence per line.

Input: The first line (N) will contain the number of data sets. Each data set consists of two lines, each line a string 60 characters long.

Output: For each data set, print out either “none” or “position X Y long”

Constraints:

1<=N<=10

Example Input file

3

```
AAAAAAAAAACCCCCCAATTCCCCCAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTTTTTTTTTTGGGGGGGTTAAGGGGGTTTTTTTTTTTTTTTTTTTTTTTTTTT
CCCCCCCCCCCCCCCCCCCCCCCCCATCGATCGATCGATCGATCGATCGATCCCC
GGGGGGGGGGGGGGGGGGGGGGGGGGGTAGCTAGCTAGCTAGCTAGCTAGCTAGGGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
```

Output to screen:

```
position 18 4 long
position 26 30 long
none
```

8. Web

Program Name: Web.java

Input File: web.dat

Your web browser keeps track of the sites you've visited, so you can use the forward and back buttons to navigate. You have to visit sites (either by typing in a URL, clicking on a hyperlink, or clicking on a bookmark/favorite) before you can go back. Before you can go forward, you must have gone back at least one site. Write a program to keep track of your forward and back buttons! Determine what website you end up browsing on, or determine if the buttons you pressed were invalid.

For this program, the first button **COULD** be invalid. It doesn't have to be the last button push. After a button push, there could be more websites typed in.

Input: The first line (N) consists of the number of data sets in the file. Each data set consists of a single integer T followed by T lines. Each subsequent line either contains a website or the letters "b" or "f". All characters will be lowercase.

Output: Print out the word "invalid" if a back or forward button is not possible. If it is valid, print out the website that the data set ends on.

Constraints:

$1 \leq N \leq 10$

$1 \leq T \leq 20$

(Continued on next page...)

(Problem 8 continued)

Example Input file

```
3
5
amazon.com
yahoo.com
apluscompsci.com
b
b
5
amazon.com
yahoo.com
apluscompsci.com
google.com
f
15
amazon.com
apluscompsci.com
apple.com
b
b
b.com
c.com
d.com
e.com
b
b
b
b
f
f
```

Output to screen:

```
amazon.com
invalid
c.com
```

Explanation of test cases:

In the 1st data set, you visit 3 sites and then go back twice to the original site, amazon.com.

In the 2nd data set, you visit 4 sites and then try to go forward, which is invalid.

In the 3rd data set, you go to 3 sites, go back to the original site, amazon.com. Then go to 4 more websites, go back 4 times to amazon.com, then go forward twice to c.com.

9. Trapped

Program Name: Trapped.java

Input File: trapped.dat

You are in the wilderness. You have a map showing your elevation in hundreds of feet. You broke your leg hiking and you cannot go down or up a steep pitch (a maximum change in 1 unit of elevation is all you can manage). Now, given a map coordinate starting point, find out whether you can hike out on your broken leg, or if you'll need helicopter rescue. A successful rescue will connect the starting point to the edge of the map (matrix) with a change in elevation of 1 unit in any direction.

Note:

1. There could be more than one rescue point, so just indicate if it is possible to rescue yourself.
2. You can move horizontally and vertically, but not diagonally.
3. Your program must NOT assume this is the only map! The judge data will be different!

Input: The first 10 lines contain the map (a 10x10 grid of the integers 1-9). The next line contains the number of data sets (N). Each subsequent line contains two integers, the coordinates of the starting point, row then column (R and C), the indices of the matrix 0-9. Each starting position will be inside the map, not on the edge (all rows/columns greater than 0 and less than 9).

Output: For each data set output either "I made it!" or "Need a chopper!" on a separate line.

Constraints:

$1 \leq N \leq 10$

$0 < R < 9$

$0 < C < 9$

Example Input file

```
2222222222
2313234312
2123551122
1332234321
2233334444
3123336665
3222364446
7777776646
6533265566
5555555546
5
1 2
2 4
7 5
6 6
8 4
```

Output to screen:

```
I made it!
Need a chopper!
I made it!
Need a chopper!
Need a chopper!
```

10. AlsoPurchased

Program Name: AlsoPurchased.java

Input File: alsopurchased.dat

When you click on an Amazon.com purchase, the website will add items to your page that are similar to what you've bought. For example, if you buy the first Harry Potter book, the rest of the series will come up as "customers also purchased...." at the bottom of the page.

This program will simulate that. Given a series of books available at Amazon.com (title, author, genre) and a title of a purchased book, find a list of recommended purchases that "customers also purchased." First priority is by author. If there are less than 3 titles by that author, then check for genre. For this program there are only 8 genres: sci-fi, fantasy, fiction, non-fiction, biography, romance, comedy, education. There may be many titles, so print out the first 3 books, first by author (alphabetical by title), then by genre (also alphabetical by title). The same book may be in different categories (author or genre) so only print the book title one time.

Input: The first 100 lines contains the books available. This will be the same in the judge data set. Each line has the form A*B*C, where A is the title, B is the author, and C is the genre, separated by an asterisk (*). These books will be in random order.
The next line contains the number of data sets (N). Each subsequent line contains the title of a book.

Output: Each data set will consist of 3-5 lines separated by a blank line. The first line of each data set will be the purchased title. The next line will be "Customers also purchased:" The next 1-3 lines will contain the appropriate book titles. Separate each data set with a blank line.

Constraints:

$1 \leq N \leq 10$

Example Input file

```
Harry Potter and the Sorcerer's Stone*J.K. Rowling*fantasy
Harry Potter and the Chamber of Secrets*J.K. Rowling*fantasy
Harry Potter and the Prisoner of Azkaban*J.K. Rowling*fantasy
Harry Potter and the Goblet of Fire*J.K. Rowling*fantasy
Harry Potter and the Order of the Phoenix*J.K. Rowling*fantasy
Harry Potter and the Half-Blood Prince*J.K. Rowling*fantasy
Harry Potter and the Deathly Hallows*J.K. Rowling*fantasy
Blue Pelican Java*Charles Cook*education
The Hunt for Red October*Tom Clancy*fiction
... /*91 more lines of input...see data file*/
3
Harry Potter and the Sorcerer's Stone
Einstein
Blue Pelican Java
```

(Continued on next page...)

(Problem 10 continued)

Output to screen:

Harry Potter and the Sorcerer's Stone
Customers also purchased:
Harry Potter and the Chamber of Secrets
Harry Potter and the Deathly Hallows
Harry Potter and the Goblet of Fire

Einstein
Customers also purchased:
A Beautiful Mind
Thomas Jefferson: The Art of Power
Truman

Blue Pelican Java
Customers also purchased:
Chemistry
Physics

11. ConVow

Program Name: ConVow.java

Input File: convow.dat

A lot of programs are about codes. Here's another one! The task is to jumble up a message by changing the consonants to vowels in a string of text. Use the series of vowels (aeiou) to replace each consonant, and then cycle through if you have more than 5 consonants. Here is an example:

```
apple pie is great
aaeie oie iu aeeai
```

You might see that this is would be hard to decipher—don't worry about that for this program!

Input: There will be an unknown number of data sets, but less than 40. Each line of text will contain only lowercase characters and spaces (no symbols).

Output: Show each line of text with no consonants.

Example Input file

```
apple pie is great
alabama jam always awesome
melekalikimaka is the thing to say
abcdefghijklmnop
```

Output to screen:

```
aaeie oie iu aeeai
aaaaeia oau aaeiao aueaoee
aaaaiaoiuiaaea ii oue aeioo uo aae
aaeieouaieiouaee
```

12. Balance

Program Name: Balance.java

Input File: balance.dat

At the end of the month, the bank sends me a statement showing how many checks I wrote, how many debit card transactions, and how many deposits I made. During the month, I always try to write down what transactions I had made, but occasionally I forget one or make a mistake. Most of the time, I reconcile my statement and I am correct. Sometimes, I find that I forgot to record a debit (I almost always remember to record a deposit). Or sometimes, I might put the debit (negative) as a credit (positive) or vice-versa. In this program, you will reconcile a bank statement to check for 3 possibilities:

- A. It is correct!
- B. The account owner forgot one entry, either a credit or debit.
- C. The account owner transposed one credit as a debit, or vice-versa.

When I record transactions, they are not always in the order received by the bank. So they could be in any order.

Input: The first line (N) consists of the number of data sets. Each data set starts with an integer T, the number of transactions recorded by the bank. After the next T lines, there will be another integer X, the number of transactions recorded by the account owner. Debits will have parentheses around the number and credits will not.

Output: For each data set there could be 5 print options:

- a) correct
- b) one entry missed: debit of ____
- c) one entry missed: credit of ____
- d) credit of ____ should be debit
- e) debit of ____ should be credit

Constraints:

$1 \leq N \leq 10$

$1 \leq T \leq 20$

$1 \leq X \leq T$

(Continued on next page...)

(Problem 12 continued)

Example Input file

```
3
5
1200.00
(20.00)
(30.00)
(99.99)
400.00
5
1200.00
(30.00)
(99.99)
400.00
(20.00)
5
1200.00
(20.00)
(30.00)
(99.99)
400.00
4
1200.00
(30.00)
(99.99)
400.00
5
1200.00
(20.00)
(30.00)
(99.99)
400.00
5
1200.00
(30.00)
99.99
400.00
(20.00)
```

Output to screen:

```
correct
one entry missed: a debit of 20.00
credit of 99.99 should be debit
```

13. Percent

Program Name: Percent.java

Input File: percent.dat

When I help my daughters with their math homework, sometimes we have problems with decimals and percents. The number 0.515 is also equal to 51.5%. For a decimal number, change it to a percent, to the nearest tenth of a percent.

Input: The first line (N) consists of the number of data sets. Each data set contains 1 double type, the decimal value, D.

Output: Print out the percent to the nearest tenth of a percent.

Constraints:

$1 \leq N \leq 10$

$0.00 \leq D < 1.00$

Example Input file

```
3
0.515
0.25
0.98989
```

Output to screen:

```
51.5%
25.0%
99.0%
```

14. DNA2

Program Name: DNA2.java

Input File: dna2.dat

In problem 7, we assumed that Chargaff's rules were observed --- that A-T and C-G pairings were correct. Sometimes there are errors in replication. When DNA replicates, there is an enzyme that makes the copy – DNA polymerase. As it copies in the 3' to 5' direction, it also checks for errors in the replication process. Write a program that simulates this!

Here is an example of palindromes:

```
AAAAAAAAAAGGGGGGAATTCCCCC AAA (strand 1)
TTTTTTTTTTCCCCCCTCAAGGGGGTTT (strand 2)
123456789012345678901234567890
```

The 19th character in the 2nd strand should be a T. Then, the palindrome is the 12th-27th letters, so position 12 and 16 long. For this program, DO NOT assume that the base pairs are correctly matched according to Chargaff's Rules (A-T and C-G). You must correct the 2nd strand if there is a base pair mis-match. Then check for palindromic DNA sequences.

If there is no palindrome, just print out the corrected sequence (or original sequence if no errors were detected).

If there was a palindrome, print out the original or corrected sequence with the position and length of the palindrome found.

Input: The first line (N) will contain the number of data sets. Each data set consists of two lines, each line a string 60 characters long.

Output: For each data set, print out the correct DNA sequence and either "none" or "position X Y long"

Constraints:

1 ≤ N ≤ 10

Example Input file

3

```
AAAAAAAAAAGGGGGGAATTCCCCCAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTTTTTTTTTCCCCCCTCAAGGGGGTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
CCCCCCCCCCCCCCCCCCCCCCCCCATCGATCGATCGATCGATCGATCGATCCCC
GAGAGAGGGCGGTGGGGGTGGGGGTAGCTAGCTAGCTAGCTAGCTAGCTAGGGGG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTCTTTTCTTTTTTTGTTTTTTGGGGTTTTTTTTTTTTTTTTTTTTTTTTTTTT
```

(Continued on next page...)

(Problem 14 continued)

Output to screen:

```
AAAAAAAAAAGGGGGGAATTCCCCCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTTTTTTTTTCCCCCCTTAAGGGGGGTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
position 12 16 long
CCCCCCCCCCCCCCCCCCCCCATCGATCGATCGATCGATCGATCGATCGATCCCCC
GGGGGGGGGGGGGGGGGGGGGGGGGGGTAGCTAGCTAGCTAGCTAGCTAGCTAGGGGG
position 26 30 long
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
none
```

15. FamilyFeud2

Program Name: FamilyFeud2.java

Input File: familyfeud2.dat

On the TV game show the Family Feud, when contestants give an answer not in the survey top answers, there is an X that shows up on the screen. You want to write a program that shows the X in a square shape. Given the size of the square, show a picture using the character capital X. The letter X will be a border on each side and along the diagonals, according to the examples below.

Input: The first line (N) consists of the number of data sets. Each data set contains one odd integer T, the size of the square “X.”

Output: For each data set, print out the T sized square with one blank line between each one.

Constraints:

$1 \leq N \leq 10$

$5 \leq T < 40$ (where $T \% 2 == 1$)

Example Input File

```
3
5
7
15
```

Output to screen:

```
XXXXXX
XX  XX
X  X  X
XX  XX
XXXXXX
```

```
XXXXXXXX
XX  XX
X  X  X  X
X  X  X
X  X  X  X
XX  XX
XXXXXXXX
```

```
XXXXXXXXXXXXXXXXXXXX
XX                      XX
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
X  X                      X  X
XX                      XX
XXXXXXXXXXXXXXXXXXXX
```