

A+ Computer Science

January 2012

Computer Science Competition Hands-On Programming Set

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.
2. All problems have a value of 60 points.
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.
5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.
6. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

Number	Name
Problem 1	Picture
Problem 2	3 Box
Problem 3	Math
Problem 4	Gum Gum for Three Three
Problem 5	Triangles
Problem 6	Book
Problem 7	Palindrome
Problem 8	Towards the Sun
Problem 9	Excel
Problem 10	OldBoy
Problem 11	Sameness
Problem 12	Digits

Good luck!

1. Picture

Program Name: Picture.java

Input File: none

General Statement : Print out the picture as shown below.

Input: none

Output: Print out the picture as shown below.

Assumptions – Helpful Hints : none

Example Input File

none

Example Output to screen:

```
#####  
##33#####33##  
#####33#####  
##33#####33##  
#####33#####  
#####33#####  
##33#####33##  
#####33#####  
##33#####33##  
#####33#####
```

2. 3 Box

Program Name: Box.java

Input File: box.dat

General Statement : Print out the 3 box as shown below.

Input: The first line in the data file will indicate the number of data sets to follow. Each data set will contain the size of the box to be printed.

Output: Print out the 3 box of the appropriate size as shown below.

Assumptions – Helpful Hints : none

Example Input File

```
3
3
5
4
```

Example Output to screen:

```
###
#3#
###
```

```
#####
#333#
#333#
#333#
#####
```

```
####
#33#
#33#
####
```

3. Math

Program Name: Math.java

Input File: math.dat

General Statement : You are on planet jj and they do math a bit differently. Math is done using the following operators : @, %, and #. @ is multiply by 3. % is add 5. # is subtract 7. That is all they can do in terms of math operations. Each expression will start with a number and then a list of operators.

Input: The first line in the data file will indicate the number of data sets to follow. Each data set will contain an expression to solve.

Output: Print out the answer formatted to 2 decimal places.

Assumptions – Helpful Hints : none

Example Input File

```
3
3 @ %
10.4 # % @
8 #
```

Example Output to screen:

```
14.00
25.20
1.00
```

4. Gum Gum for Three Three

Program Name: Gum.java

Input File: gum.dat

General Statement : You are lost in the museum and keep walking by a giant rock head that says “gum gum for three three” each time you walk by. Print out the number of times you have walked by the giant rock head after reading in the data file.

Input: The data file will contain an unknown number of lines.

Output: Print out the number of lines in the data file.

Assumptions – Helpful Hints : none

Example Input File

```
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
gum gum for three three
```

Example Output to screen:

```
11
```

5. Triangles

Program Name: Triangles.java

Input File: triangles.dat

General Statement : Read in a letter and a number. The number indicates how big the letter triangle should be. The number indicating the size of the triangle will have a range from 0 to 250. $\text{num} \geq 0$ and $\text{num} \leq 250$

Input: The first number indicates the number of data sets to follow. Each data set will contain one letter and one number. All letter input will be uppercase.

Output: Print out the appropriate sized letter triangle as shown below.

Assumptions – Helpful Hints : The letters must wrap around from Z to A. If you start with Z and have to print 5 levels, you must wrap around and start with A after the Z level is complete.

Example Input File

```
3
5 A
3 Z
4 C
```

Example Output to screen:

```
  A
 BB
CCC
 DDDD
EEEE
```

```
  Z
 AA
 BBB
```

```
   C
  DD
 EEE
FFFF
```

6. Book

Program Name: Book.java

Input File: none

General Statement : Print out the word Book as shown below.

Input: none

Output: Print out the word Book as shown below.

Assumptions – Helpful Hints : none

Example Input File

none

Example Output to screen:

```
#####  
# # # # #  
#####  
# # # # #  
#####  
# # # # #  
#####
```

7. Palindrome

Program Name: Pal.java

Input File: pal.dat

Palindrome is a string that is read the same forwards and backwards. Non-palindromic strings are useless. For a string S, a string T is a subsequence of S if and only if T can be made by deleting 0 or more characters from S. Given an input string, we want to purge all useless characters. In other words, we want to find the biggest palindrome that is a subsequence of the input string. For example, if the input string is “GGABZXVYCCDBA”. Then the largest palindrome subsequence is “ABCCBA”.

Input

The first line in the data files represents the number of data sets to follow. Each data set contains a single line containing a string of length no greater than 100.

Output

Print the size of the largest palindrome subsequence.

Example Input File

```
3
GGABZXVYCCDBA
ABGBAD
ABC
```

Example Output To Screen

```
6
5
1
```


8. Towards the Sun

Program Name: Sun.java

Input File: sun.dat

Generally a plant likes to move towards the sun. Otherwise it dies and becomes useless. At the current point in time, a plant starts off somewhere in a two dimensional square grid. Since the sun is located in the southeast corner. The plant would like to move as far south and as far east as possible in terms of Manhattan distance. In other words, every time the plant moves 1 step to the south or to the east, it gains a unit of sunlight, and every time the plant moves 1 step to the north or the west, it loses a unit of sunlight. Output the maximum gain in sunlight the plant can achieve.

Input

The first line in the data file is an integer that represents the number of data sets to follow. Each data set will contain several lines. The first line contains two integers R and C denoting the number of rows and columns in the grid. The next R line contains a string of length C where a “#” represents a wall in the grid and “P” is the starting location of the plant

Output

For each input print the maximum gain in sunlight.

Example Input File

```
2
4 5
#####
#P  #
##  ##
#####
3 10
#####
#  P  #  #
#####
```

Example Output To Screen

```
2
3
```

9. Excel

Program Name: Excel.java

Input File: excel.dat

Excel columns are not numbered with numeric digits. They are useless. Instead, they are labeled with letters. Column 1 = A, Column 2 = B, Column 26 = Z, Column 27 = AA, Column 28 = AB, and so on. In other words, the first 26 columns are labeled A through Z. The next 26^2 columns are labeled AA through ZZ. The next 26^3 columns are labeled AAA through ZZZ and so on. Output the n-th excel column label.

Input

The first line in the data file is an integer that represents the number of data sets to follow. Each data set contains the column number as an integer.

Output

Print the label for each input column number

Example Input File

```
5
26
27
345
703
1321321
```

Example Output To Screen

```
Z
AA
MG
AAA
BWDPA
```

10. OldBoy

Program Name: Oldboy.java

Input File: oldboy.dat

You are given two strings S1 and S2. Both strings contain some number of letters (possibly zero) and exactly one asterisk. You must replace the asterisks with a sequence of letters (possibly the empty string) such that the resulting two strings are equal. There may be infinitely many solutions, but you must return the shortest possible resulting string. If there is no such solution, print “USELESS”.

Input

The first line in the data file is an integer that represents the number of data sets to follow. Each data set contains one line. Each line contains two strings S1 and S2.

Output

For each input print the shortest such string. If no such string exists, print “USELESS”.

Example Input File

```
5
HELL*WORLD HELLO*RLD
HELLO* HI*
GOOD*BYE *
*SAMPLECASE ONESAMPLE*
GOOD* *GOOD
```

Example Output To Screen

```
HELLOWORLD
USELESS
GOODBYE
ONESAMPLECASE
GOOD
```

11. Sameness

Program Name: Same.java

Input File: same.dat

Some words are the same. Some words are different. Then there are words that are what we call “same difference”. Two strings are same different if the letters in one word can be remapped to get the second word. The ordering of the letters remains unchanged. No two letters may map to the same letter, but a letter may map to itself. For example the words “abca” and “zbxz” are same different because we can map ‘a’ to ‘z’, ‘b’ to ‘b’ and ‘c’ to ‘x’. Given 2 strings S1 and S2, find whether they are same different or not.

Input

The first line in the data file is an integer that represents the number of data sets to follow. Each data set consists of a single line containing S1 and S2.

Output

For each input, print “SAME” if they are same different or “USELESS” if they are not.

Example Input File

```
4
abca zbxz
zbxz opqr
ab aa
aa cc
```

Example Output To Screen

```
SAME
USELESS
USELESS
SAME
```

12. Digits

Program Name: Digits.java

Input File: digits.dat

If two integers differ by one digit, they are called a smooth pair. For example, 128 and 28 are different by 1 digit. 73 and 74 differ by 1 digit. 2040 and 40 differ by 1 digit. This is because we pad the smaller number with leading zeros when comparing digits. Given a number K, find the smallest possible non-negative number M, such that N and M are a smooth pair.

Input

The first line in the data file is an integer that represents the number of data sets to follow. Each input consists of a single integer K where K is less than 2,000,000,000.

Output

For each input print the smallest possible non-negative number M where (N,M) is a smooth pair.

Example Input File

```
4
9
900123
3000
1907654321
```

Example Output To Screen

```
0
123
0
907654321
```