



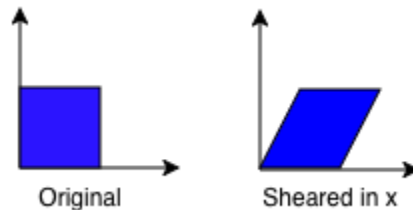
38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #1: Shear Transformation

Background Information: In computer graphics, a shear transformation is a mapping that displaces each point in a fixed direction by an amount proportional to its signed distance from a given line parallel to that direction. For a point (x, y) , a transformation in the x direction by a shear amount s returns a new point $(x + sy, y)$.



Given a point (x, y) and a shear value s , calculate the new point based on the shear transformation formula.

Programming Problem:

Input: Integers x , y , and s all on separate lines. Each integer input is in the interval $[-1000, 1000]$.

Output: The new point based on the formula $(x + sy, y)$, where x and y are separated by a single space.

Example 1:	Input:	Output:
	2	2 1
	1	
	0	

Example 2:	Input:	Output:
	3	4 -1
	-1	
	-1	

Example 3:	Input:	Output:
	-2	1 3
	3	
	1	



38th Annual High School Programming Contest

sponsored by **transfinder**

April 17, 2026

Green Problem #2: Can You Run for Governor?

Background Information: Your friend is a U.S. citizen and is considering running for governor. Is she eligible? Essentially, all states require a person to be a U.S. citizen. However, states have different minimum age requirements. The following table lists the state name, the two letter postal abbreviation, and the minimum age for each state. Note, that Vermont has no minimum age.

Alabama AL 30	Alaska AK 30	Arizona AZ 25	Arkansas AR 30	California CA 18	Colorado CO 30	Connecticut CT 30	Delaware DE 30	Florida FL 30	Georgia GA 30
Hawaii HI 30	Idaho ID 30	Illinois IL 25	Indiana IN 30	Iowa IA 30	Kansas KS 25	Kentucky KY 30	Louisiana LA 30	Maine ME 30	Maryland MD 30
Massachusetts MA 25	Michigan MI 30	Minnesota MN 25	Mississippi MS 30	Missouri MO 30	Montana MT 30	Nebraska NE 30	Nevada NV 25	New Hamps NH 30	New Jersey NJ 30
New Mexico NM 30	New York NY 30	N Carolina NC 30	N Dakota ND 30	Ohio OH 18	Oklahoma OK 31	Oregon OR 30	Pennsylvania PA 30	Rhode Is. RI 18	S Carolina SC 30
South Dakota SD 21	Tennessee TN 30	Texas TX 30	Utah UT 30	Vermont VT 0	Virginia VA 30	Washington WA 18	West Virginia WV 30	Wisconsin WI 18	Wyoming WY 30

Programming Problem:

Input: The two letter postal abbreviation for a state on one line, both letters must be uppercase.
On the next input line will be the age of a potential candidate. This must be a non-negative integer.

Output: The word ELIGIBLE, if the candidate is eligible to be Governor of the specified state with the given age; otherwise NOT ELIGIBLE.

Example 1: Input: Output:
 AL
 30
 ELIGIBLE

Example 2: Input: Output:
 NH
 10
 NOT ELIGIBLE

Example 3: Input: Output:
 VERMONT
 1
 ELIGIBLE



38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #3: Too abundant, too deficient, or just perfect!

Background Information: A positive integer N is said to be a *perfect number* if the sum of the proper factors of N is equal to N . A proper factor of N is a positive integer less than N that divides N . For example, the proper factors of 12 are 1, 2, 3, 4, and 6. If the sum of the proper factors of N is greater than N , N is an *abundant number*. For example, 12 is an abundant number because $1+2+3+4+6 = 16$ which is greater than 12. If the sum of the proper factors is less than N , N is a *deficient number*. Ten is an example of a deficient number because the proper factors of 10 are 1, 2, and 5 and $1+2+5 = 8 < 10$. The smallest perfect number is 6. Six is a perfect number because the proper factors are 1, 2, and 3 and $1+2+3 = 6$. Note that 1 is a deficient number.

Write a program that counts the number of perfect, abundant, and deficient numbers in the inclusive interval A to B .

Programming Problem:

Input: A pair of positive integers A and B with $A \leq B \leq 10,000$. The A and B will be input on two lines.

Output: The values of A and B followed by the number of perfect, abundant, and deficient numbers A and B inclusive. These five integers must be output on five consecutive lines.

Example 1 :	Input:	1	Output:	1
		250		250
				2
				57
				191

Example 2:	Input:	1000	Output:	1000
		2000		2000
				0
				248
				753



38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #4: FizzBuzzWhizz

Background Information: The Fizz Buzz Whizz problem is an extension of the classic FizzBuzz problem, where a program would have to replace all integers within a range that are divisible by 3 with the word “Fizz”, all integers within that same range that are divisible by 5 with “Buzz” and all integers within that same range that are divisible by both 3 and 5 with “FizzBuzz”. So, in FizzBuzz, the sequence

11 12 13 14 15 16 17 18 19 20

would now become

11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz

In this problem, you will be given positive integers A and B which will define the range of integers, starting from A and ending with B . You will then be given three positive integers, x , y , and z . Replace all integers within the range that are divisible by x with Fizz, replace all integers within the range that are divisible by y with Buzz, replace all integers within the range that are divisible by both x and y with FizzBuzz, or replace all integers within the range that are divisible by x , y , and z with Whizz. If an integer in the range is not divisible by x , y , or z the integer is not replaced. Print out the new range with replacements to the screen on one line with one space between values.

Programming Problem:

Input: positive integers A , B where $A \leq B \leq 100$, followed by three positive integers x , y and z . Each input value will be on its own line.

Output: The single Fizz-Buzz-Whizz line with one space between words and numbers.

Example 1: Input:

1
10
2
5
1

Output:

1 Fizz 3 Fizz Buzz Fizz 7 Fizz 9 Whizz

Example 2: Input:

13
22
31
127
2

Output:

13 14 15 16 17 18 19 20 21 22

(more examples on the next page)

Example 3: Input:

5
15
4
6
17

Output:

5 Buzz 7 Fizz 9 10 11 FizzBuzz 13 14 15

Example 4: Input:

10
20
2
3
4

Output:

Fizz 11 Whizz 13 Fizz Buzz Fizz 17 FizzBuzz 19 Fizz

Example 5: Input:

17
17
17
17
17

Output:

Whizz



38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #5: Transfinder Nameplates

Background Information: The Director of Transportation for Geegleburg School District wants to purchase two nameplates for each district bus driver, one for the driver's bus and one for a board in the bus garage.

For this programming problem, each driver will have a unique first name.

The transportation director picks up the set of nameplates from the purchasing office and is carrying them across the parking lot when he hears a clinking sound. Unfortunately, he has dropped one of the nameplates into a sewage drain.

But fortunately he has an enthusiastic team of programmers who can write a program that inputs the names on the tags that have not gone down the drain and it outputs the name of the one that did.

It will be the name that only occurs once in the input.

Programming Problem:

Input: Positive odd integer $K \leq 1000$, followed by K names in uppercase

Output: The name N which appears exactly once

Example 1: Input:

1 KAREN

Output:

KAREN

Example 2: Input:

3 ROBIN ROBIN ROBYN

Output:

ROBYN

Example 3: Input:

5 BILL JOE JOE BILLY BILL

Output:

BILLY



38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #6: An AI Bot's Look And Say

Background Information: Starting with a seed number s , a Look-and-Say sequence $LS(s)$ is a sequence of integers generated by an AI bot from a starting seed number. The AI bot produces the sequence by recording in text exactly what it states using its voice-enabled functionality when reading off the digits. The AI bot is talented and avoids repetition. For example, when it produces part of a sequence when the seed number has repeating digits the bot just says the number of those digits. For example when the bot sees 333333 it will say "six threes" which then becomes 63 in the sequence.

Here are three easy examples:

1 has "one 1" in it, so $LS(1) = 11$.

11 has "two 1s" in it, so $LS(11) = 21$.

21 has "one 2 and one 1" in it, so $LS(21) = 1211$.

Write a program that will input the seed value s , and a non-negative integer n . Your output will be the result of applying the Look-and-Say function $LS(x)$ n times, first on s , and each subsequent time on the result of the previous function application.

For example, if you input 1 for s and 4 for n your program will compute:

$LS(1) = 11$ and then $LS(11) = 21$ and then $LS(21) = 1211$ and then $LS(1211) = 111221$

Another example, if you input 37 for s and 5 for n your program will compute:

$LS(37) = 1317$ and then $LS(1317) = 11131117$ and then $LS(11131117) = 31133117$ and then

$LS(31133117) = 1321232117$ and finally $LS(1321232117) = 111312111213122117$

Programming Problem:

Input: The seed number s and the number of function applications n , on separate lines.

Output: The result of applying the function n times, starting on s .

Example 1: Input: 1
 4
 Output: 111221

Example 2: Input: 22
 10
 Output: 22

Example 3: Input: 5
 5
 Output: 1113122115

Example 4: Input: 3
 2
 Output: 1113

Example 5: Input: 37
 5
 Output: 111312111213122117



38th Annual High School Programming Contest

sponsored by transfinder

April 17, 2026

Green Problem #7: Sock It To Me! Sock It To Me! Sock It To Me!

Background Information: You are setting out socks from your sock drawer in sequences of length $n \geq 3$. The socks are either black or white. Since you are familiar with mathematics and especially the binary representation of positive integers, you know there are 2^n different sequences of length n that can be made with black and white socks. How many of those sequences do not have three consecutive white socks?

Your program must find the number of sequences made from all sequences of input n number of socks that do not have three white socks in a row.

Programming Problem:

Input: positive integer $3 \leq n \leq 70$

Output: The number of possible sequences of socks of length n , that do not have three consecutive white socks.

Example 1: Input:
 3

 Output:
 7

Example 2: Input:
 4

 Output:
 13

Example 3: Input:
 6

 Output:
 44

Example 4: Input:
 60

 Output:
 1383410902447554