

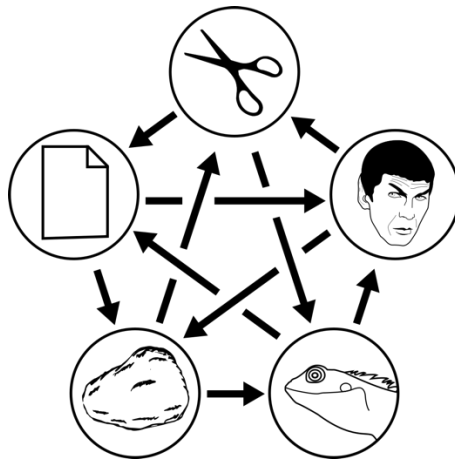
# 2021 All Ireland Programming Competition

## Selection Round

Authors: Andrew Nash, Bogdan Marcu, Sabin Tabirca

Graphics and Proofread: Aine Curran

### P1: Rock Paper Scissors Lizard Spock



We are given the signs made by two players in a game known as Rock Paper Scissors Lizard Spock, and want to inform the players of the outcome (who won or lost). Each player chooses a sign that they will play:

Rock(R)      Paper(P)      sCissors(C)      Lizard(L)      Spock(S)

The rules of the game are defined as:

- Scissors cuts Paper
- Paper covers Rock
- Rock crushes Lizard
- Lizard poisons Spock
- Spock smashes Scissors
- Scissors decapitates Lizard
- Lizard eats Paper
- Paper disproves Spock
- Spock vaporizes Rock
- Rock crushes Scissors

**Input:** Two lines (representing two players), each containing one letter (one of R,P,C,L,S) representing the sign that each player chose to play

**Output:** A single character - 1 if the first player (first line) won, 2 if the second player (second line) won or the character 'D' in case of a draw

Case	Inputs	Outputs
1.	R P	2
2.	C P	1
3.	L L	D

## P2 :Brackets

4 Valid brackets

**()() => 4**

4 valid brackets 1 invalid bracket

**()))() => 4**

We are given a sequence of opening and closing brackets. We consider any sequence of brackets to be *valid* if every opening ( is closed by a ), and every closing ) is opened by a ( i.e. (()), ()() are valid, while ((), )) are not.

We want to find the length of the maximum valid subsequence of a sequence of brackets.

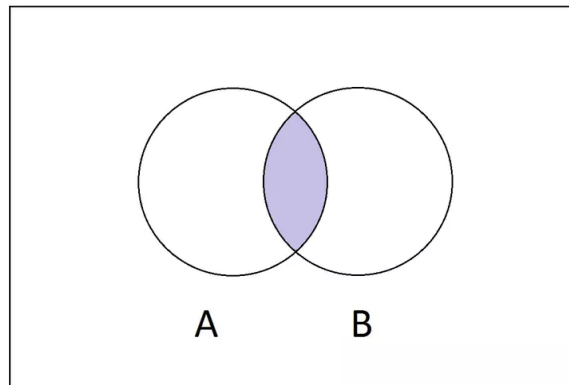
These subsequences do not necessarily have to be continuous, i.e. ()() is a subsequence of (()))().

**Input:** A single integer n denoting the length of the bracket sequence, followed by the sequence on a new line

**Output:** A single integer m, the length of the maximal subsequence

Case	Inputs	Outputs
1.	3 (()	2
2.	7 ((()))()	6

## P3: Set Intersection



We want to write a program that will help find the intersection of two sets. A set is a sequence of distinct integer elements  $\{2, 1, 3, 4\}$  is a set, while  $\{1, 1, 0\}$  is not a set as 1 appears twice. We are guaranteed that the two sets are non-empty. An integer is an element of the intersection of two sets if it appears in both individual sets.

**Input:** The input has 4 lines. The first line represents the number of elements in the first set. The second line contains the elements of the first set separated by a space “ ”. The third line gives the number of elements for the second set and the fourth one gives the elements of the second set.

**Output:** One line containing the elements of the intersection (in increasing order), or the string “Empty Set” if the intersection is empty.

### Examples:

Case	Inputs	Outputs
1	4 2 1 -1 100 3 2 1 3	1 2

2	3  1 3 5  5  0 2 4 6 8	Empty Set
---	--	-----------

## P4: Palindromes in Primes

```

2
30203
903020309
3790302030973
98379030203097389
969837903020309738969
9996983790302030973896999
72999698379030203097389699927
997299969837903020309738969992799
9099729996983790302030973896999279909
94909972999698379030203097389699927990949
779490997299969837903020309738969992799094977
7977949099729996983790302030973896999279909497797
17797794909972999698379030203097389699927990949779771
751779779490997299969837903020309738969992799094977977157
7375177977949099729996983790302030973896999279909497797715737
72737517797794909972999698379030203097389699927990949779771573727
987273751779779490997299969837903020309738969992799094977977157372789
3098727375177977949099729996983790302030973896999279909497797715737278903
70309872737517797794909972999698379030203097389699927990949779771573727890307
397030987273751779779490997299969837903020309738969992799094977977157372789030793
3539703098727375177977949099729996983790302030973896999279909497797715737278903079353
36353970309872737517797794909972999698379030203097389699927990949779771573727890307935363
333635397030987273751779779490997299969837903020309738969992799094977977157372789030793536333
3433363539703098727375177977949099729996983790302030973896999279909497797715737278903079353633343
99343336353970309872737517797794909972999698379030203097389699927990949779771573727890307935363334399

```

We are given a single integer  $n$ . Find the largest prime factor of  $n$  that is also a palindrome. Reminder: the prime factors are all of the distinct prime numbers that divide  $n$  without remainder. A number is a palindrome if its digits are the same when read forwards, or backwards, e.g 1221, 222, and 3

**Input:** A single integer  $n$

**Output:** A single integer, the maximum prime palindromic factor, or -1 if no such factor exists

Case	Inputs	Outputs
1	1746997	353
2	329	7
3	19	-1

**Explanation:**

For case 1 :  $1746997 = 353 \cdot 101 \cdot 7 \cdot 7$

## P5: Haunted Manor



We, the players, are given a grid representing a haunted manor filled with mirrors, empty cells, and cells containing different types of monsters. Players must determine the type and position of each monster:

The different types of monsters are:

- V: Vampire, who can be seen directly, but not inside a mirror
- Z: Zombie, who can be seen both directly and in a mirror.
- G: Ghost, who cannot be seen directly but can be seen in a mirror

Each cell on the border of the grid has a window through which you can peer into the manor. For each border, you are given the number of monsters visible through that window. Line of sight will bounce off the mirrors, making it possible or impossible to see certain monsters.

There are two types of mirror:

- \: Diagonal down
- /: Diagonal up

The manor is always a square grid. None of the cells are empty.

Example: This 3 by 3 manor has 0 vampires, 4 zombies and 2 ghosts. 3 mirrors are present.

0	/	\	/	2
1				1
3				3

No monster can be seen from the top because of the mirrors. The middle row seems to contain two ghosts since only 1 monster is visible from either side. Using all the reported sightings of monsters through the windows, we can easily come to the configuration below.

0	/	\	/	2
1	G	G	Z	1
3	Z	Z	Z	3

**Input:** Each on a new line:

The number of each of the types of monsters separated by spaces: nr\_vampies nr\_zombies nr\_ghosts

The length and width of the grid separated by spaces: nr\_rows nr\_cols

The sightings for the top edge from left to right

The sightings for the bottom edge from left to right

The sightings for the left edge from top to bottom

The sightings for the right edge from top to bottom

Next nr\_rows lines containing nr\_cols characters. (.) containing one of the monsters, \ or / for a mirror

**Output:** The layout of the grid with the . replaced by the letter representing the monster

Case	Inputs	Outputs
1.	<pre> 0 4 2 3 3 0 0 0 3 3 2 0 1 3 2 1 3 /\ ... ... </pre>	<pre> /\ GGZ ZZZ </pre>
2.	<pre> 2 3 4 4 4 3 3 3 0 1 1 0 0 3 1 3 1 0 3 2 0 \.\ .../ .../. ...\ </pre>	<pre> \Z\ GGZ\ GV/V Z\G/ </pre>