

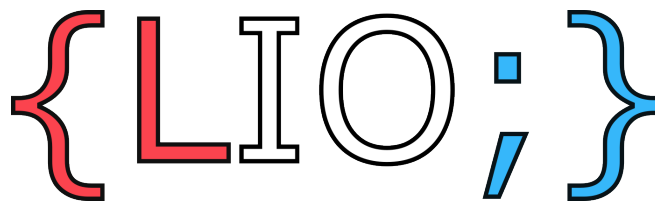
Lëtzebuerger Informatiksolympiad 2024

Qualification Round

Task descriptions

Instructions

- The allowed programming languages are Python 3, Java and C/C++.
- All the programs must be realized in the form of a console application. For instructions how to realize a console application in the allowed programming languages, please refer to the remarks on the site www.infosolympiad.lu under the heading *The tasks*.
- Under the input of the program is meant either the direct entry of data from the keyboard or the redirection from a text file in console mode. Under output of the program is meant either the direct display of data to the screen or the redirection to a text file in console mode.
- The formats of the input and output data shown in the execution examples must absolutely be respected.
- For testing, submitting and evaluating a program, the source file with a file extension `py`, `java` or `c/cpp` must be uploaded to the automated online judge CMS (Contest Management System), accessible via the homepage www.infosolympiad.lu or directly via the URL <http://158.64.46.20:81>. Please use your personal login (username and password) to access your account on the CMS. The filename of the single source file should be the same than the task name. Please refer to the CMS for technical details on how to test and submit a program.
- Please refer to the CMS for technical details like time limits and memory limits as well as compilation commands.
- You have the right to ask questions via the CMS, but the answers will not teach you how to use a programming language nor tell you how to solve the tasks by using a specific algorithm. The questions should be in relation with the CMS or should treat clarification issues concerning the task descriptions.



The Connect Four game

Description

This task is based on the popular 2-players Connect Four game, also known as *Connect 4*, *Four Up*, *Plot Four*, *Find Four* etc. For more information about this game, please refer to Wikipedia. In this task, we will not implement the game as it is played (i.e. alternatively piling up the tokens from the bottom), but we simply consider that all tokens (of red or yellow color) are already placed in a square grid of size $N \times N$ and that we must simply find out if there are at least 4 connected tokens of one color in a row, a column or diagonally. The number of tokens of each color is irrelevant in this task. The result of the program depends on the layout of the tokens. If there are exclusively at least 4 connected tokens of one color, the output is that color. If there are at least 4 connected tokens for both red and yellow, or if there are no 4 connected tokens of one color at all, the output is NOPE (see execution examples).

Task

Based on the input data, write a program which determines the correct output according to the token layout.

Constraints

- $N \in \mathbb{N}, 4 \leq N \leq 20$

Input and output of program

Input data

The first line indicates the square grid size N .

The next N lines indicate the colors of the tokens in the grid, row by row. The upper-case letter R stands for a red token, the letter Y stands for a yellow token

Output data

The text RED, YELLOW or NOPE, according to the token layout

Execution example

Input 1

```
6
RYYYRY
RYRRYY
YRYRYY
RRYRRR
RYRRYY
YRRYYR
```

Output 1

```
RED
```

Input 2

```
5
YRYRY
RYRYY
YYRYY
RYYR
YRYRY
```

Output 2

```
NOPE
```

Input 3

```
5
YRYRY
YYRYY
RRYRR
YYYYY
RRRRR
```

Output 3

```
NOPE
```

Input 4

```
7
RRYYRYR
YRYRYRY
YRYYRY
RYRYYR
RYYRRYY
YRRYYRY
YRRRRYR
```

Output 4

```
NOPE
```

Indeed,

- In example 1, there are four R in column 4
- In example 2, there no four connected tokens of any color
- In example 3, the bottom row contains five aligned R, the second to last row contains five Y and there also are diagonals containing four consecutive Y
- In example 4, there are four R in the last row but also five Y starting row 2 column 3 and going down diagonally

Technical constraints

Task name	connect4
Input file	standard input
Output file	standard input
Time limit	1 second
Memory limit	256 megabytes

Remark

The program does not need to check if the input data is valid (data types, values and format), we suppose that it always is.

Squares and Cubes

Description

Leo recently learned about square and cube numbers. (Recall that n is a square, if $n = a^2$ for some integer a , and that n is a cube, if $n = b^3$ for some integer b .)

Here are some square numbers Leo found: 9, 25 and 81. And here are some cubes he found: 8, 27 and 125.

Leo got a bit bored of just finding squares or cubes, so he wants to find all numbers less than N which are both squares and cubes.

An example of such a number is $729 = 27^2 = 9^3$.

Task

Write a program which outputs all numbers less than N , which are both squares and cubes.

Constraints

- $1 \leq N \leq 10^8$

Input and output of program

Input data

The first line contains an integer N .

Output data

One line containing all numbers less than N which are both squares and cubes (in ascending order) separated by spaces.

Execution example

Input

70

Output

1 64

Distribution of points

Subtask	Points	Constraints/Description
1	10	$N < 1000$
2	15	No additional constraints

Technical constraints

Task name	squarecubes
Input file	standard input
Output file	standard input
Time limit	0.1 second
Memory limit	256 megabytes

Letzflix

Description

Leo recently learned about the streaming platform Letzflix. To watch movies on Letzflix, you need to buy a subscription costing C euros which lasts K days. (That is for the next K days after buying a subscription you can watch any movies you want). If you buy a subscription on day i , it is valid on the day of buying, until day $i + K$ inclusive.

Leo is very organised, so he already has a schedule where he planned out when he wants to watch N movies. He knows he wants to watch the i th movie on the day a_i .

However Leo doesn't like wasting money, so he wants you to help him minimise the amount of money he spends on subscriptions, while still watching all movies on his schedule.

Task

Write a program which calculates the minimum amount of money Leo needs to spend on Letzflix subscriptions, keeping to his schedule.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq C \leq 100$
- $0 \leq K \leq 10^9$
- $1 \leq a_i \leq 10^9$

Input and output of program

Input data

The first line contains three integers separated by a space: N (the number of movies), C (the cost of the subscription), K (the length of the subscription).

The second line contains N integers a_i describing the day Leo wants to watch the i th movie.

Output data

Output a single integer: the minimum amount of money Leo has to spend on Letzflix subscriptions.

Execution example

Input

```
5 2 4
1 3 6 11 13
```

Output

```
6
```

Explanation of Example

Leo needs to spend 6 euros on subscriptions on day 1, 6 and 11.

Distribution of points

Subtask	Points	Constraints/Description
1	5	$K = 10^9$
2	5	$K = 0$ (The subscription is only valid on the day you buy it)
2	15	No additional constraints

Technical constraints

Task name	letzflix
Input file	standard input
Output file	standard input
Time limit	1 second
Memory limit	256 megabytes

Goldbach's Conjecture

Description

One of the oldest unproven problems in mathematics, Goldbach's Conjecture, states that every even natural number N greater than 2 can be written as the sum of two prime numbers, which we will refer as **pairs** of primes that sum to N . However, the conjecture does not specify how many such pairs exists for a certain N , this peaks your interest and you decide to count all these pairs.

Task

Given an even positive integer $N > 2$, print the amount of pairs of primes that sum to N .

Example

The number 22 is an even positive integer that can be written as the sum $3 + 19$, $5 + 17$ and $11 + 11$ and therefore has 3 different pairs.

Constraints

- $2 < N \leq 10^7$, N even

Input and output of program

Input data

The first line contains N .

Output data

Your program should print on a single line the amount of pairs of primes for N .

Execution example

Input

22

Output

3

Distribution of points

Subtask	Points	Constraints/Description
1	10	$N < 10000$
1	10	$N < 100000$
2	5	No additional constraints

Technical constraints

Task name	goldbach
Input file	standard input
Output file	standard input
Time limit	1 second
Memory limit	256 megabytes