lab 02 - 11.03.’24

1)

The program:

Your program must find the missing digit in a single line of sudoku.
As a reminder, in a sudoku grid, each line must contain all digits from 1 to 9, one time each.

INPUT:

One line of a sudoku grid with a missing digit marked by a '?'

OUTPUT:

The missing digit.

EXAMPLE:

Input	Output
837?9514	6

2)

Terry is trying to eat those N candies and he can put up to K candies in his mouth at once.
What is the fastest way for Terry to eat his candies ?

For example, let's say Terry wants to eat 7 candies and that he can put in his mouth up to 3 candies :
The fastest way for him to eat those candies is to eat 3, 3 and then 1 candies.

The output should then be :
3 3 1

Input

Two integers separated by a space: N the number of candies, K the size of Terry's mouth.

Output

The fastest way for Terry to eat his candies.
A single line of integer separated by spaces: The possibility, i.e. the numbers of candy that Terry puts in his mouth at each step.

Constraints

N won't exceed 10000.
K won't exceed 1000.

Example

Input	Output
1 1	1

3)

01.Reverse A String

```
A = "the sky is blue"
 "blue is sky the"
```

4)

02.FizzBuzz

Given a positive integer A, return an array of strings with all the integers from 1 to N.
But for multiples of 3 the array should have “Fizz” instead of the number.
For the multiples of 5, the array should have “Buzz” instead of the number.
For numbers which are multiple of 3 and 5 both, the array should have “FizzBuzz” instead of the number.
Look at the example for more details.

A = 5
Return: [1 2 Fizz 4 Buzz]

5)

03.Smaller or equal elements

Given an sorted array A of size N. Find number of elements which are less than or equal to B.

NOTE: Expected Time Complexity O(log N)

Input Format

First agument is an integer array A of size N.
Second argument is an integer B.

Output Format
Return an integer denoting the number of elements which are less than or equal to B.
input:
A = [1, 3, 4, 4, 6]
B = 4
output:
4

6)

Given a regular `n`-gon, find the measures of a single interior angle, a single exterior angle, and if the angles are both **odd** or **even**.

The sum of the exterior angles is always 360.

Find the measures of the interior angles using the following formula: (`n`-2)*180/`n`

Input

Line 1: a number `n` for number of sides the polygon has.

Output

Line 1: space-separated integers, interior angle measure `i`, and exterior angle measure `e`.
Line 2: output **even** if both the angle measures are even, otherwise **odd**

Constraints

0 < `n` < 40
`n` is also chosen such that the angles are always integers and don't need any rounding.

Example

Input	Output
3	60 120 even

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("interior angle, exterior angle")

```

7)

The game mode is **REVERSE** : You do not have access to the statement. You have to guess what to do by observing the following set of tests:

01 Test 1

Input	Expected output
string	6 gnirts

02 Test 2

Input	Expected output
variable	8 elbairav

03 Test 3

Input	Expected output
pointer	7 retniop

04 Test 4

Input	Expected output
function	8 noitcnuf

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

_input = input()

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("answer")

```

8)

Output the (signed) size of the range [N , 1000- N]

Input

The integer N

Output

one integer

Constraints

0 ≤ N ≤ 1000

Example

Input	Output
15	970

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())

```

9)

Your program must output a hollow square composed of the `#` symbol with sides of length `N`.

INPUT:
`N`, an integer.

OUTPUT:
A `N` x `N` square over `N` lines made of `#` symbols.

CONSTRAINTS:
0 < `N` < 100

EXAMPLE:

Input	Output
5	##### # # # # # # #####

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("#answer")

```

10)

Print a word containing `n` times the `n`th letter of the alphabet (uppercase).

Input

`n`: a number

Output

`word` : a word containing `n` times the `n`th letter of the alphabet (uppercase)

Constraints

1<= `n` <= 26

Example

Input	Output
5	EEEEE

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())

```

11)

The game mode is **REVERSE** : You do not have access to the statement. You have to guess what to do by observing the following set of tests:

01 Test 1

Input	Expected output
What I saw was completely out of the ordinary.	saw was

02 Test 2

Input	Expected output
I am stressed, need some desserts!	stressed desserts

03 Test 3

Input	Expected output
Step carefully, I have snakes as pets.	Step pets

04 Test 4

Input	Expected output
A reverse spelling of palindromes.	<missing>

05 Test 5

Input	Expected output
One.	<missing>

06 Test 6

Input	Expected output
Mom of my mom is my grandmom.	<missing>

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

s = input()

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("R")

```

12)

Given an integer N, you must calculate i raised to the power N.

i or the imaginary unit is equal to the square root of -1. Thus, i^2 = -1, i^3 = -i, i^4 = 1 ...
Note: powers of i are cyclical.

Input

Line 1: An integer N, which represents the power to which i is raised to.

Output

Line 1: What i^N is equivalent to.

Constraints

0 ≤ N ≤ 100000

Example

Input	Output
1	i

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("answer")

```

13)

A hunter goes hunting ducks (represented by `D`) and woodcocks (represented by `W`).
He has `n` number of shots and uses all of them during the hunt.

The sequence of shots from the start of the hunt to the end is represented by a string of length `n` (every character is the result of a shot): every character of the string is either:
- `D` meaning a bird of type duck has been killed;
- `W` meaning a bird of type woodcock has been killed;
- `f` meaning no bird has been killed;
- `*` meaning two birds (with the same type of the last bird killed) have been killed;

It is guaranteed that whenever a `*` appears in the string of shots at least one bird has been already killed.

At the end of the hunt, the hunter sells all the birds of the type with most birds killed and keeps for himself the other kills. If the number of birds of each type is the same, he sells all the birds of woodcock type and keeps for himself the rest.
Your goal is to find out the number of birds the hunter keeps for himself.

Input

Line 1: an integer `n` which is the number of shots
Line 2: a string `shots` which is the sequence of the results of the shots from the first shot (first character) to the last shot (last character)

Output

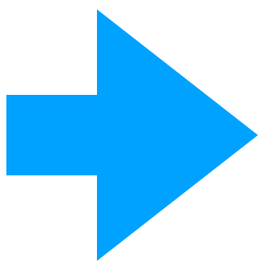
An integer `k` which is the number of birds the hunter keeps for himself

Constraints

$2 \leq n \leq 20$
$0 \leq k \leq 40$

Example

Input	Output
10 WWWWWWWDDD	3



```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())
shots = input()

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("answer")

```

14)

At school, children learn to write letter by letter. They first learn to write A, then B, then C, etc. (both in upper and lower case). Given a list of names, who will be able to write their name first ?
In case of equality, print the names in alphabetic order separated by spaces

Input

Line 1: `n` the number of names
`n` **next lines :** each `name`, containing only letters

Output

The names that would be learned first, separated by spaces

Constraints

1<=`n`<100
1<=length(`firstname`)<=20

Example

Input

```
6
Clement
Julien
Sebastien
Alexandre
Pierre
Christophe
```

Output

```
Pierre
```

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())
for i in range(n):
    firstname = input()

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("firstname")

```

15)

Sunrise. You drive your bus to the first bus stop without any passenger.
At each bus stop, some people get in the bus, and other people get off the bus.
You have to count how many people are in the bus at the end, including yourself.

Input

First line : the number `n` of bus stops.
n following lines : two numbers `getIn` and `getOff`

Output

Only one integer, `nbPeople`, the number of people in the bus at the end.

Constraints

`n` >= 1
`getIn` >= 0
`getOff` >= 0

Example

Input

```
3
2 0
4 1
1 3
```

Output

```
4
```

```python
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.

n = int(input())
for i in range(n):
    get_in, get_off = [int(j) for j in input().split()]

# Write an answer using print
# To debug: print("Debug messages...", file=sys.stderr, flush=True)

print("nbPeople")
```