CODILITY TASK EXAMPLES – C, C++, C#

Programming language: C#   Spoken language: English

A non-empty array A consisting of N numbers is given. The array is sorted in non-decreasing order. The *absolute distinct count* of this array is the number of distinct absolute values among the elements of the array.

For example, consider array A such that:

```
A[0] = -5
A[1] = -3
A[2] = -1
A[3] =  0
A[4] =  3
A[5] =  6
```

The absolute distinct count of this array is 5, because there are 5 distinct absolute values among the elements of this array, namely 0, 1, 3, 5 and 6.

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given a non-empty array A consisting of N numbers, returns absolute distinct count of array A.

For example, given array A such that:

```
A[0] = -5
A[1] = -3
A[2] = -1
A[3] =  0
A[4] =  3
A[5] =  6
```

the function should return 5, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647];
- array A is sorted in non-decreasing order.

Programming language: C   Spoken language: English

There are N coins, each showing either heads or tails. We would like all the coins to form a sequence of alternating heads and tails. What is the minimum number of coins that must be reversed to achieve this?

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given an array A consisting of N integers representing the coins, returns the minimum number of coins that must be reversed. Consecutive elements of array A represent consecutive coins and contain either a 0 (heads) or a 1 (tails).

Examples:

1. Given array A = [1, 0, 1, 0, 1, 1], the function should return 1. After reversing the sixth coin, we achieve an alternating sequence of coins [1, 0, 1, 0, 1, 0].

2. Given array A = [1, 1, 0, 1, 1], the function should return 2. After reversing the first and fifth coins, we achieve an alternating sequence [0, 1, 0, 1, 0].

3. Given array A = [0, 1, 0], the function should return 0. The sequence of coins is already alternating.

4. Given array A = [0, 1, 1, 0], the function should return 2. We can reverse the first and second coins to get the sequence: [1, 0, 1, 0].

Assume that:

- N is an integer within the range [1..100];
- each element of array A is an integer that can have one of the following values: 0, 1.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

CODILITY TASK EXAMPLE – COBOL

Programming language: [Cobol ▾]   Spoken language: [English ▾]

You are given a COBOL module containing a field INPUT-STRING of 3000 characters.

Your task is to change the given implementation, so it will perform the following steps:

1. Search INPUT-STRING for every occurrence of such substring: 'NUM' followed by at least three consecutive digits.
2. Count the sum of all the 3-digit numbers from the substrings found in point 1.
3. If no occurrences of NUM followed by at least three digits are found, return 0.
4. If the sum exceeds 9999, return 9999.
5. The returned sum should be stored in the OUTPUT-RESULT field.

Input and output examples:

INPUT: 'abcNUM099cdeNUM111NUMNUM3333abcabc' OUTPUT: 543

INPUT: 'abcNUM 99abcabcNUM11NUM1234567890' OUTPUT: 123

INPUT: 'NUMNUMNUM12abcabcNUM NUM1NUMabcNUM1a8abcabc' OUTPUT: 0

Hints

- Please do not create any additional fields in the LINKAGE-SECTION, as this will corrupt the tests.
- In WORKING-STORAGE section don't use VALUE clause when you intend to change this value later.

nab

CODILITY TASK EXAMPLES - DART

Your task is to implement an Iterable over Fridays

Requirements

Create a function named `fridaysIterator` which

- takes `date of first Friday` (String) in `yyyy-mm-dd` format as an argument;
- returns `iterable` (Iterable), where each yielded item is the date of the next Friday in `yyyy-mm-dd` format, and the first yielded item is a date given as an argument.

Assumptions

- Assume that the function always gets a valid date in `yyyy-mm-dd` (String) format as an argument.
- Assume that the function always gets the date of a Friday as an argument.

Hints

- Remember about leap years and that months have different numbers of days.
- Your solution will be evaluated based on its correctness; performance and coding style will not be assessed.

Available packages/libraries

- Dart 2.13.3

Examples

```
final Iterable it = fridaysIterator('2021-06-04');
final Iterator iterator = it.iterator;

iterator.moveNext()
print(iterator.current); // '2021-06-04'

iterator.moveNext()
print(iterator.current); // '2021-06-11'

iterator.moveNext()
print(iterator.current); // '2021-06-18'

iterator.moveNext()
print(iterator.current); // '2021-06-25'

iterator.moveNext()
print(iterator.current); // '2021-07-02'
```

Your task is to implement a Dart Extension on Int.

Requirements

Write an Extension on Int named `MovieExtension`* containing three methods (each of which should return a *String*):

1. `toMovieLength`* which converts minutes to hours and minutes. If the movie is less than an hour long, only minutes should be returned. If the movie is representable by full hours only, only hours should be returned. Example: *154 => "2h 34min"*
2. `toPrice`* which gets a currency (String) as a required argument and converts the movie price to a number with two decimal places and a currency symbol. Example: *2342 => "23.42$"*
3. `toAvailableCopies`* which converts the number of available copies to an appropriate message:
   - if there are more than 15 copies: "*num* left"**
   - if there are 15 or fewer copies: "Only *num* left!"**
   - if there are no copies: "We ran out of copies!".

*The class and method names must not be changed.

***num* is the number of available copies, e.g. "40 left", "Only 3 left!"

Assumptions

- You can assume that method `toMovieLength` is always called on a movie length represented by minutes greater than 0.
- You can assume that methods `toPrice` and `toAvailableCopies` are always called on a number greater than or equal to 0.

Hints

- Your solution will be evaluated based on its correctness; performance and coding style will not be assessed.

Available packages/libraries

- Dart 2.13

Examples

```
234.toMovieLength() == "3h 54min"
60.toMovieLength() == "1h"
32.toMovieLength() == "32min"

530.toPrice("$") == "5.30$"
42.toPrice("PLN") == "0.42PLN"
0.toPrice("$") == "Free"

35.toAvailableCopies() == "35 left"
2.toAvailableCopies() == "Only 2 left!"
0.toAvailableCopies() == "We ran out of copies!"
```

nab

CODILITY TASK EXAMPLES - GO

Programming language: Visual Basic ▾   Spoken language: English ▾

Write a function solution that, given an integer N, returns the maximum possible value obtained by inserting one '5' digit inside the decimal representation of integer N.

Examples:

1. Given N = 268, the function should return 5268.

2. Given N = 670, the function should return 6750.

3. Given N = 0, the function should return 50.

4. Given N = –999, the function should return –5999.

Assume that:

- N is an integer within the range [–8,000..8,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Programming language: Go ▾   Spoken language: English ▾

A non-empty array A consisting of N integers is given. The *amplitude* of this array is defined as the largest possible difference between two of its elements, i.e.:

$$amplitude(A) = \max\{ A[P] - A[Q] : 0 \le P, Q < N \}$$

Write a function:

```
func Solution(A []int) int
```

that, given a non-empty array A consisting of N integers, returns its amplitude.

For example, given array A such that:

```
A[0] = 10
A[1] = 2
A[2] = 44
A[3] = 15
A[4] = 39
A[5] = 20
```

the function should return 42.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..1,000,000];
- each element of array A is an integer within the range [0..5,000,000].

nab

CODILITY TASK EXAMPLES - JAVA

Programming language: Java 8 ▾   **Spoken language:** English ▾

A rectangle is called *rectilinear* if its edges are all parallel to coordinate axes. Such a rectangle can be described by specifying the coordinates of its bottom-left and top-right corners.

Write a function

```
class Solution { public int solution(int K, int L, int M, int N, int P, int Q, int R, int S); }
```

that, given eight integers representing two rectilinear rectangles (one with lower-left corner (K, L) and upper-right corner (M, N), and another with lower-left corner (P, Q) and upper-right corner (R, S)), returns the area of their intersection. The function should return 0 if the intersection is a point or a line segment, or if the rectangles do not intersect. The function should return −1 if the area of the intersection exceeds 2,147,483,647.

For example, given integers:

```
K = 0      L = 2      M = 5      N = 10
P = 3      Q = 1      R = 20     S = 15
```

the function should return 16.

The intersection of the two rectangles is a rectilinear rectangle whose lower-left corner is (3,2) and upper-right corner is (5,10), and its area equals 16.

Write an **efficient** algorithm for the following assumptions:

- K, L, M, N, P, Q, R and S are integers within the range [−2,147,483,648..2,147,483,647].

Programming language: Java 8 ▾   **Spoken language:** English ▾

A string is a *palindrome* if it has exactly the same sequence of characters when read left-to-right as it has when read right-to-left. For example, the following strings are palindromes:

- "kayak",
- "codilitytilidoc",
- "neveroddoreven".

A string A is an *anagram* of a string B if A can be obtained from B by rearranging the characters. For example, in each of the following pairs one string is an anagram of the other:

- "mary" and "army",
- "rocketboys" and "octobersky",
- "codility" and "codility".

Write a function:

```
class Solution { public int solution(String S); }
```

that, given a non-empty string S consisting of N characters, returns 1 if S is an anagram of some palindrome and returns 0 otherwise.

For example, given S = "dooernedeevrvn", the function should return 1, because "dooernedeevrvn" is an anagram of the palindrome "neveroddoreven". Given S = "aabcba", the function should return 0, because there is no palindrome such that S is an anagram of it.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- string S consists only of lowercase letters (a–z).

nab

CODILITY TASK EXAMPLES - JAVASCRIPT

Programming language: JavaScript ▾

Consider an array A of N integers. Indices of this array are integers from 0 to N−1. Take an index K. Index J is called an *ascender* of K if A[J] > A[K]. Note that if A[K] is a maximal value in the array A, then K has no ascenders.

Ascender J of K is called *the closest ascender* of K if abs(K−J) is the smallest possible value (that is, if the distance between J and K is minimal). Note that K can have at most two closest ascenders: one smaller and one larger than K.

For example, let us consider the following array A:

```
A[0] = 4    A[1]  = 3    A[2]  = 1
A[3] = 4    A[4]  = -1   A[5]  = 2
A[6] = 1    A[7]  = 5    A[8]  = 7
```

If K = 3 then K has two ascenders: 7 and 8. Its closest ascender is 7 and distance between K and 7 equals abs(K−7) = 4.

Write a function:

```
class Solution { public int[] solution(int[] A); }
```

that, given an array A of N integers, returns an array R of N integers, such that (for K = 0,..., N−1):

- if K has the closest ascender J, then R[K] = abs(K−J); that is, R[K] is equal to the distance between J and K,
- if K has no ascenders then R[K] = 0.

For example, given the following array A:

```
A[0] = 4    A[1]  = 3    A[2]  = 1
A[3] = 4    A[4]  = -1   A[5]  = 2
A[6] = 1    A[7]  = 5    A[8]  = 7
```

the function should return the following array R:

```
R[0] = 7    R[1]  = 1    R[2]  = 1
R[3] = 4    R[4]  = 1    R[5]  = 2
R[6] = 1    R[7]  = 1    R[8]  = 0
```

Result array should be returned as an array of integers.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..50,000];
- each element of array A is an integer within the range [−1,000,000,000..1,000,000,000].

Programming language: Java 8 ▾   Spoken language: English ▾

You are a programmer in a scientific team doing research into particles. As an experiment, you have measured the position of a single particle in N equally distributed moments of time. The measurement made in moment K is recorded in array A as A[K].

Now, your job is to count all the periods of time when the movement of the particle was stable. Those are the periods during which the particle doesn't change its velocity: i.e. the difference between any two consecutive position measurements remains the same. Note that you need at least three measurements to be sure that the particle didn't change its velocity.

For example:

```
1,  3,  5,  7,  9  is stable (velocity is 2)
    7,  7,  7,  7  is stable (particle stays in place)
    3, -1, -5, -9  is stable (velocity is -4)
            0,  1  is not stable (you need at least three measurements)
1,  1,  2,  5,  7  is not stable (velocity changes between measurements)
```

More formally, your task is to find all the periods of time A[P], A[P+1], ..., A[Q] (of length at least 3) during which the movement of the particle is stable. Note that some periods of time might be contained in others (see example test).

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given array A consisting of N integers representing the results of the measurements, returns the number of periods of time when the movement of the particle was stable. The function should return −1 if the result exceeds 1,000,000,000.

Examples:

Given array A = [−1, 1, 3, 3, 3, 2, 3, 2, 1, 0] the function should return 5, because there are five periods during which the movement of the particle is stable, namely: (0, 2), (2, 4), (6, 9), (6, 8) and (7, 9). Note that the last two periods are contained by (6, 9).
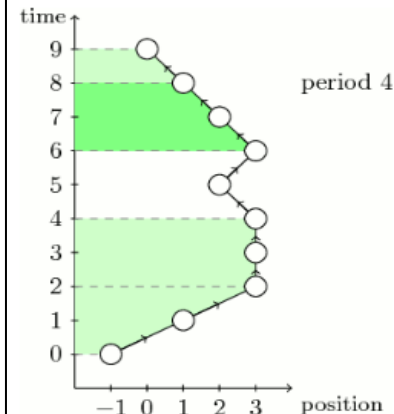
Assume that:

- N is an integer within the range [0..100];
- each element of array A is an integer within the range [−1,000,000,000..1,000,000,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

nab

CODILITY TASK EXAMPLES - KOTLIN

Left panel:

Programming language: Kotlin

An integer X and a non-empty array A consisting of N integers are given. We are interested in which elements of A are equal to X and which are different from X. Let's mark all elements of A equal to X with green color and the rest (elements different from X) with red color. Our goal is to split array A into two parts (left and right), such that the number of green elements in the left part is the same as the number of red elements in the right part. More formally, we are looking for a number K such that:

- $0 \leq K \leq N$, and
- the number of elements equal to X in A[0..K−1] is the same as the number of elements different from X in A[K..N−1]. (For K = 0, A[0..K−1] does not contain any elements. For K = N, A[K..N−1] does not contain any elements.)

Write a function:

```
fun solution(X: Int, A: IntArray): Int
```

that, given an integer X and a non-empty array A consisting of N integers, returns the value of number K satisfying the above conditions. It can be shown such number K always exists and it is unique.

For example, given integer X = 5 and array A such that:

```
A[0] = 5
A[1] = 5
A[2] = 1
A[3] = 7
A[4] = 2
A[5] = 3
A[6] = 5
```

your function should return 4.

We will mark with green color A[0], A[1] and A[6]. The rest (A[2], A[3], A[4] and A[5]) will be marked with red color.

Your function should return 4, because:

- two of the elements of A[0..3] are green (equal to X), namely A[0] = A[1] = X, and
- two of the elements of A[4..6] are red (different from X), namely A[4] and A[5].

In other example, given X = 7 and A = [6, 5, 4, 3, 2, 1], your function should return 6.

Every element will be red.

Your function should return 6, because:

- no elements of A[0..5] are green (equal to X), and
- there are no elements in the right part, so there are no red elements also.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- X is an integer within the range [0..100,000];
- each element of array A is an integer within the range [0..100,000].

Right panel:

Programming language: Kotlin   Spoken language: English

A non-empty array A consisting of N integers is given.

Array A represents a linked list. A list is constructed from this array as follows:

- the first node (the head) is located at index 0;
- the value of a node located at index K is A[K];
- if the value of a node is −1 then it is the last node of the list;
- otherwise, the successor of a node located at index K is located at index A[K] (you can assume that A[K] is a valid index, that is 0 ≤ A[K] < N).

For example, for array A such that:

```
A[0] =  1
A[1] =  4
A[2] = -1
A[3] =  3
A[4] =  2
```
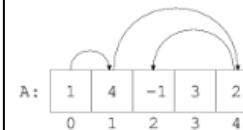

the following list is constructed:

- the first node (the head) is located at index 0 and has a value of 1;
- the second node is located at index 1 and has a value of 4;
- the third node is located at index 4 and has a value of 2;
- the fourth node is located at index 2 and has a value of −1.

Write a function:

```
fun solution(A: IntArray): Int
```

that, given a non-empty array A consisting of N integers, returns the length of the list constructed from A in the above manner.

For example, given array A such that:

```
A[0] =  1
A[1] =  4
A[2] = -1
A[3] =  3
A[4] =  2
```

the function should return 4, as explained in the example above.

Assume that:

- N is an integer within the range [1..200,000];
- each element of array A is an integer within the range [−1..N−1];
- it will always be possible to construct the list and its length will be finite.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

nab

CODILITY TASK EXAMPLES - LUA

Programming language: [Lua ▾]

You are given a non-empty string S consisting of N characters. In this problem we consider only strings that consist of lower-case English letters (a-z) and spaces.

S can be divided into *words* by splitting it at the spaces and removing them. We want to reverse every word in S.

For example, given S = "we test coders", there are three words: "we", "test" and "coders". Reversing the words gives "ew", "tset" and "sredoc".

The goal is to return a string with every word in string S reversed and separated by spaces, so the result in the above example should be "ew tset sredoc". You can assume that if there are K spaces in S then there are exactly K + 1 words.

Write a function:

    function solution(S)

that, given a non-empty string S consisting of N characters, returns the reversal of every word of S.

For example, given S = "we test coders", the function should return "ew tset sredoc", as explained above.

Write an **efficient** algorithm for the following assumptions:

- the length of string S is within the range [1..200,000];
- string S consists only of lower-case letters (a-z) and spaces.

Task idea contributed by Stephen Law of Electrum.

Programming language: [Lua ▾]

An integer P is a *whole square* if it is a square of some integer Q; i.e. if $P = Q^2$.

Write a function:

    function solution(A, B)

that, given two integers A and B, returns the number of whole squares within the interval [A..B] (both ends included).

For example, given A = 4 and B = 17, the function should return 3, because there are three squares of integers in the interval [4..17], namely $4 = 2^2$, $9 = 3^2$ and $16 = 4^2$.

Assume that:

- A and B are integers within the range [−10,000..10,000];
- A ≤ B.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

nab

CODILITY TASK EXAMPLES – OBJECTIVE C

Programming language: Objective-C ✔

Four integers A, B, C and D are given. A *shuffle* of them is any array S consisting of these four integers in some order. If all the given integers are unique, then there are 24 different shuffles of them.

The *best* shuffle of the given integers is any shuffle S of them such that the value of:

```
F(S) = abs(S[0]-S[1]) + abs(S[1]-S[2]) + abs(S[2]-S[3])
```

is maximal.

Write a function:

```
    int solution(int A, int B, int C, int D);
```

that, given four integers A, B, C and D, finds their best shuffle S and returns the value of F(S).

For example, consider the following integers:

```
  A = 5     B = 3     C = -1     D = 5
```

The best shuffle of them is as follows:

```
  S[0] =  5
  S[1] = -1
  S[2] =  5
  S[3] =  3
```

and the result is F(S) = 14.

Assume that:

- A, B, C and D are integers within the range [−1,000,000..1,000,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Programming language: Objective-C ✔

A *binary gap* within a positive integer N is any maximal sequence of consecutive zeros that is surrounded by ones at both ends in the binary representation of N.

For example, number 9 has binary representation 1001 and contains a binary gap of length 2. The number 529 has binary representation 1000010001 and contains two binary gaps: one of length 4 and one of length 3. The number 20 has binary representation 10100 and contains one binary gap of length 1. The number 15 has binary representation 1111 and has no binary gaps. The number 32 has binary representation 100000 and has no binary gaps.

Write a function:

```
    int solution(int N);
```

that, given a positive integer N, returns the length of its longest binary gap. The function should return 0 if N doesn't contain a binary gap.

For example, given N = 1041 the function should return 5, because N has binary representation 10000010001 and so its longest binary gap is of length 5. Given N = 32 the function should return 0, because N has binary representation '100000' and thus no binary gaps.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..2,147,483,647].

nab

CODILITY TASK EXAMPLES - PASCAL

Programming language: Pascal ▼

You are given a string S consisting of letters 'a' and/or 'b'. A block is a consecutive fragment of S composed of the same letters and surrounded by different letters or string endings. For example, S = "abbabbaaa" has five blocks: "a", "bb", "a", "bb" and "aaa".

What is the minimum number of additional letters needed to obtain a string containing blocks of equal lengths? Letters can be added only at the beginning or at the end of an existing block.

Write a function:

```
function solution(S: PChar): longint;
```

that, given a string S of length N, returns the minimum number of additional letters needed to obtain a string containing blocks of equal lengths.

Examples:

1. Given S = "babaa", the function should return 3. There are four blocks: "b", "a", "b", "aa". One letter each should be added to the first, second and third blocks, therefore obtaining a string "bbaabbaa", in which every block is of equal length.

2. Given S = "bbbab", the function should return 4. Two letters each should be added to the second and third blocks, therefore obtaining a string "bbbaaabbb", in which every block is of equal length.

3. Given S = "bbbaaabbb", the function should return 0. All blocks are already of equal lengths.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..40,000];
- string S consists only of the characters "a" and/or "b".

Programming language: Pascal ▼

A string S consisting of N characters is considered to be *properly nested* if any of the following conditions is true:

- S is empty;
- S has the form "(U)" or "[U]" or "{U}" where U is a properly nested string;
- S has the form "VW" where V and W are properly nested strings.

For example, the string "{[()()]}" is properly nested but "([)()]" is not.

Write a function:

```
class Solution { public int solution(String S); }
```

that, given a string S consisting of N characters, returns 1 if S is properly nested and 0 otherwise.

For example, given S = "{[()()]}", the function should return 1 and given S = "([)()]", the function should return 0, as explained above.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [0..200,000];
- string S consists only of the following characters: "(", "{", "[", "]", "}" and/or ")".

nab

CODILITY TASK EXAMPLES - PERL

Programming language: Perl

A non-empty array A consisting of N integers and sorted in a non-decreasing order (i.e. A[0] ≤ A[1] ≤ ... ≤ A[N−1]) is given. The *leader* of this array is the value that occurs in more than half of the elements of A.

You are given an implementation of a function:

```perl
sub solution { my (@A)=@_; ... }
```

that, given a non-empty array A consisting of N integers, sorted in a non-decreasing order, returns the leader of array A. The function should return −1 if array A does not contain a leader.

For example, given array A consisting of ten elements such that:

```
A[0] = 2
A[1] = 2
A[2] = 2
A[3] = 2
A[4] = 2
A[5] = 3
A[6] = 4
A[7] = 4
A[8] = 4
A[9] = 6
```

the function should return −1, because the value that occurs most frequently in the array, 2, occurs five times, and 5 is not more than half of 10.

Given array A consisting of five elements such that:

```
A[0] = 1
A[1] = 1
A[2] = 1
A[3] = 1
A[4] = 50
```

the function should return 1.

The attached code is still **incorrect** for some inputs. Despite the error(s), the code may produce a correct answer for the example test cases. The goal of the exercise is to find and fix the bug(s) in the implementation. You can modify at most **three** lines.

Assume that:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [0..2,147,483,647];
- array A is sorted in non-decreasing order.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Programming language: Perl

Write a function:

```perl
sub solution { my ($S1, $S2)=@_; ... }
```

that, given two strings S1 and S2, returns:

- −1 if string S1 precedes string S2 in alphabetical order (disregarding lower/upper case);
- 0 if string S1 is equal to string S2 (disregarding lower/upper case);
- 1 if string S1 follows string S2 in alphabetical order (disregarding lower/upper case).

A *prefix* of a string S is any leading contiguous part of S. For example, "", "c" and "cod" are prefixes of the string "codility". If S1 is a prefix of S2 then S1 precedes string S2, if S2 is a prefix of S1 then S1 follows string S2.

For example, given S1 = "abc" and S2 = "AbA" the function should return 1.
Given S1 = "a" and S2 = "aa" the function should return −1.
Given S1 = "abc" and S2 = "Abc" the function should return 0.

Write an **efficient** algorithm for the following assumptions:

- lengths of strings S1 and S2 are within the range [0..1,000,000];
- strings S1 and S2 consist only of letters (a−z and/or A−Z).

If you know of a library function that implements exactly this functionality, please state this fact in a comment, but do not use the function.

nab

CODILITY TASK EXAMPLES - PHP

Programming language: PHP ▾

A *central sphere* of radius R is the set of all points (x, y, z) in 3D space that satisfy the following equation:

$$x^2 + y^2 + z^2 = R^2$$

We say that a set of central spheres *covers* a set of points if each of the points belongs to one of the spheres.

Assume that the following declarations are given:

```
class Point3D {
    public int x;
    public int y;
    public int z;
}
```

Write a function:

```
class Solution { public int solution(Point3D[] A); }
```

that, given an array describing of a set of points in 3D space, returns the minimum number of central spheres required to cover them.

For example, given the following array:

```
A[0].x =  0  A[0].y =  5  A[0].z =  4
A[1].x =  0  A[1].y =  0  A[1].z = -3
A[2].x = -2  A[2].y =  1  A[2].z = -6
A[3].x =  1  A[3].y = -2  A[3].z =  2
A[4].x =  1  A[4].y =  1  A[4].z =  1
A[5].x =  4  A[5].y = -4  A[5].z =  3
```

the function should return 3, because three central spheres are required to cover these points:

- a central sphere of radius **sqrt(3)** covers array element A[4],
- a central sphere of radius 3 covers array elements A[1] and A[3],
- a central sphere of radius **sqrt(41)** covers array elements A[0], A[2] and A[5].

It is impossible to cover these points with fewer central spheres.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..100,000];
- the coordinates of each point in array A are integers within the range [−10,000..10,000].

Programming language: PHP ▾

Two positive integers N and M are given. Integer N represents the number of chocolates arranged in a circle, numbered from 0 to N − 1.

You start to eat the chocolates. After eating a chocolate you leave only a wrapper.

You begin with eating chocolate number 0. Then you omit the next M − 1 chocolates or wrappers on the circle, and eat the following one.

More precisely, if you ate chocolate number X, then you will next eat the chocolate with number (X + M) modulo N (remainder of division).

You stop eating when you encounter an empty wrapper.

For example, given integers N = 10 and M = 4. You will eat the following chocolates: 0, 4, 8, 2, 6.

The goal is to count the number of chocolates that you will eat, following the above rules.

Write a function:

```
function solution($N, $M);
```

that, given two positive integers N and M, returns the number of chocolates that you will eat.

For example, given integers N = 10 and M = 4. the function should return 5, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N and M are integers within the range [1..1,000,000,000].

nab

CODILITY TASK EXAMPLES - PYTHON

Programming language: Python ▼

Write a function:

```
class Solution { public int solution(int[] A, int[] B); }
```

that, given a non-empty array A of N non-negative integers and a non-empty array B of M non-negative integers, returns the minimal value that occurs in both arrays. If there is no such value, the function should return −1.

For example, given arrays A and B such that:

```
A[0] = 1     B[0] = 4
A[1] = 3     B[1] = 2
A[2] = 2     B[2] = 5
A[3] = 1     B[3] = 3
             B[4] = 2
```

your function should return 2, since 2 is the minimal value which occurs in both arrays A and B (another value which occurs in both arrays is 3).

Given arrays A and B such that:

```
A[0] = 2     B[0] = 3
A[1] = 1     B[1] = 3
```

your function should return −1, since there is no value that occurs in both arrays.

Assume that:

- N and M are integers within the range [1..1,000];
- each element of arrays A and B is an integer within the range [0..3,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Task idea contributed by Stephen Law of Electrum.

Programming language: Python ▼

Write a function:

```
def solution(A)
```

that, given an array A consisting of N integers, returns the value (or one of the values) that occurs most often in this array.

For example, given array A such that:

```
A[0] = 20
A[1] = 10
A[2] = 30
A[3] = 30
A[4] = 40
A[5] = 10
```

the function may return 10 or 30.

Assume that:

- N is an integer within the range [1..1,000];
- each element of array A is an integer within the range [0..10,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

nab

CODILITY TASK EXAMPLES - R

Programming language: [R ▾]

You are given a data set called *biopics.csv* containing information on biographical movies. Your task is to perform some manipulations on the biopics data.

Data overview

The original biopics data come as part of the `fivethirtyeight` R package. You will be working with a preprocessed version, available for you at `./data/biopics.csv`. It contains the following columns:

- `title` – the movie's title;
- `country` – the country of production;
- `year_release` – the year the movie was released;
- `box_office` – the movie's earning at the box office, in US Dollars;
- `type_of_subject` – the occupation of the movie's subject, or the reason for their recognition;
- `lead_actor_actress` – the name of the actor or actress who played the subject.

Instructions

Write a function named `process_data()` that takes no arguments. The function should load the biopics data (this has been implemented for you), perform data manipulations described below and return a data frame with manipulated data.

Clean up the biopics data as follows:

- Filter out duplicated rows;
- Rename the variable called `box_office` to `earnings`;
- Filter out any rows for which `earnings` are missing (i.e. they are `NA`);
- Retain only movies released in 1990 or later;
- Convert the type of `type_of_subject` and `country` to factors;
- Create a new variable called `lead_actor_actress_known` that is `FALSE` if `lead_actor_actress` is `NA` and `TRUE` otherwise;
- Update `earnings` such that they are expressed in millions of dollars, instead of dollars;
- Reorder the columns in the data frame such that they are in the following order: `title`, `year_release`, `earnings`, `country`, `type_of_subject`, `lead_actor_actress`, `lead_actor_actress_known`;
- Sort the rows in descending order by `earnings`.

On top of the base R, you can make use of any function from the `dplyr` package.

If you would like to access CSV data sets locally you can download zipped files.

Programming language: [R ▾]

Write a function `calculate_rmse()` that, given numeric vectors `predicted` and `observed` (both of length `n`) returns Root Mean Square Error of these values. For every `i`, the value `predicted[i]` corresponds to `observed[i]`. Your result will be accepted if its absolute error is at most 0.0001.

The Root Mean Square Error is given by the following formula:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

where:

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values

$y_1, y_2, \ldots, y_n$ are observed values

n is the number of observations

For example, given

```
predicted <- c(4, 25, 0.75, 11)
observed <- c(3, 21, -1.25, 13)
```

the function should return 2.5.

Assume that:

- N is an integer within the range [1..1,000];
- each element in predicted and observed arrays is a value within the range [-10^9..10^9].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Apart from base R, you can use any package from `tidyverse` collection.

nab

CODILITY TASK EXAMPLES - RUBY

Programming language: Ruby

An array A consisting of N integers is given. We visit the indexes of the array in the following way. In the first step we visit the index 0; in every subsequent step we move from the visited index K to the index:

```
M = K + A[K]
```

provided M is within the array bounds. Otherwise, K is the last index visited.

Write a function:

```
def solution(a)
```

that, given an array A, returns the number of indexes that cannot be visited by the described procedure.
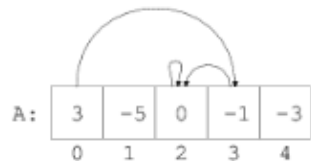
For example, for the array:

```
A[0] = 1    A[1] = 2    A[2] = 3
```


only index 2 cannot be visited, so the answer is 1.

For the array:

```
A[0] = 3    A[1] = -5    A[2] = 0    A[3] = -1    A[4] = -3
```


indexes 1 and 4 cannot be visited, so the answer is 2.

Assume that:

- N is an integer within the range [0..100];
- each element of array A is an integer within the range [−1,000,000..1,000,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Programming language: Ruby

A string S consisting of uppercase English letters is given. In one move we can delete seven letters from S, forming the word "BALLOON" (one 'B', one 'A', two 'L's, two 'O's and one 'N'), and leave a shorter word in S. If the remaining letters in the shortened S are sufficient to allow another instance of the word "BALLOON" to be removed, next move can be done. What is the maximum number of such moves that we can apply to S?

Write a function:

```
class Solution { public int solution(String S); }
```

that, given a string S of length N, returns the maximum number of moves that can be applied.

Examples:

1. Given S = "BAONXXOLL", the function should return 1.

2. Given S = "BAOOLLNNOLOLGBAX", the function should return 2.

3. Given S = "QAWABAWONL", the function should return 0.

4. Given S = "ONLABLABLOON", the function should return 1.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..200,000];
- string S consists only of uppercase letters (A–Z).

nab

CODILITY TASK EXAMPLES - SCALA

An integer M and a non-empty array A consisting of N non-negative integers are given. All integers in array A are less than or equal to M.

A pair of integers (P, Q), such that 0 ≤ P ≤ Q < N, is called a *slice* of array A. The slice consists of the elements A[P], A[P + 1], ..., A[Q]. A *distinct slice* is a slice consisting of only unique numbers. That is, no individual number occurs more than once in the slice.

For example, consider integer M = 6 and array A such that:

```
A[0] = 3
A[1] = 4
A[2] = 5
A[3] = 5
A[4] = 2
```

There are exactly nine distinct slices: (0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (2, 2), (3, 3), (3, 4) and (4, 4).

The goal is to calculate the number of distinct slices.

Write a function:

```
object Solution { def solution(m: Int, a: Array[Int]): Int }
```

that, given an integer M and a non-empty array A consisting of N integers, returns the number of distinct slices.

If the number of distinct slices is greater than 1,000,000,000, the function should return 1,000,000,000.

For example, given integer M = 6 and array A such that:

```
A[0] = 3
A[1] = 4
A[2] = 5
A[3] = 5
A[4] = 2
```

the function should return 9, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- M is an integer within the range [0..100,000];
- each element of array A is an integer within the range [0..M].

A positive integer D is a *factor* of a positive integer N if there exists an integer M such that N = D * M.

For example, 6 is a factor of 24, because M = 4 satisfies the above condition (24 = 6 * 4).

Write a function:

```
object Solution { def solution(n: Int): Int }
```

that, given a positive integer N, returns the number of its factors.

For example, given N = 24, the function should return 8, because 24 has 8 factors, namely 1, 2, 3, 4, 6, 8, 12, 24. There are no other factors of 24.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..2,147,483,647].

nab

CODILITY TASK EXAMPLES - SQL

Programming language: SQL (PostgreSQL) ⌄   Spoken language: English ⌄

You are presented with a table of data detailing people queueing for an elevator. All the people want to enter the elevator, but it can only hold a maximum weight of 1000 lbs. Your task is to find the last person who will fit without exceeding the weight restriction.

You are given a non-empty table line with the structure described by the query:

```
create table line (
    id int not null PRIMARY KEY,
    name varchar(255) not null,
    weight int not null,
    turn int unique not null,
    check (weight > 0)
);
```

where the column turn represents the position of each person in the line. Positions are numbered from 1.

For example, given:

```
id | name              | weight | turn |
---+-------------------+--------+------+
 5 | George Washington |    250 |    1 |
 4 | Thomas Jefferson  |    175 |    5 |
 3 | John Adams        |    350 |    2 |
 6 | Thomas Jefferson  |    400 |    3 |
 1 | James Elephant    |    500 |    6 |
 2 | Will Johnliams    |    200 |    4 |
```

your query should return a table containing exactly one record:

```
+------------------+
| Thomas Jefferson |
+------------------+
```

as the first three people will fit (in the order: George Washington, John Adams, Thomas Jefferson). The sum of their weight is 250 + 350 + 400 = 1000.

Note that it is possible for two rows to contain the same name, but different id. In this case, they are considered to represent two different people.

Assume that:

- the turn column contains all integers from 1 to N;
- the id column consists all integers from 1 to N;
- the first person always fits into the elevator.

Programming language: SQL (SQLite) ⌄   Spoken language: English ⌄

Given a table events with the following structure:

```
create table events (
    event_type integer not null,
    value integer not null,
    time timestamp not null,
    unique(event_type, time)
);
```

write an SQL query that, for each event_type that has been registered more than once, returns the difference between the latest (i.e. the most recent in terms of time) and the second latest value. The table should be ordered by event_type (in ascending order).

For example, given the following data:

```
event_type | value | time
-----------+-------+--------------------
2          | 5     | 2015-05-09 12:42:00
4          | -42   | 2015-05-09 13:19:57
2          | 2     | 2015-05-09 14:48:30
2          | 7     | 2015-05-09 12:54:39
3          | 16    | 2015-05-09 13:19:57
3          | 20    | 2015-05-09 15:01:09
```

your query should return the following rowset:

```
event_type | value
-----------+----------
2          | -5
3          | 4
```

For the event_type 2, the latest value is 2 and the second latest value is 7, so the difference between them is -5.

The names of the columns in the rowset don't matter, but their order does.

CODILITY TASK EXAMPLES – SWIFT

A non-empty array A consisting of N integers is given. The array consists only of integers in the range [0..N−1]. Each element of the array can be treated as a pointer to another element of the array: if A[K] = M then element A[K] points to A[M].

The array defines a sequence of jumps of a pawn as follows:

- initially, the pawn is located at position 0;
- on each jump the pawn moves from its current position K to A[K];
- the pawn jumps forever.

Since the number of possible positions of the pawn is finite, eventually, after some initial sequence of jumps, the pawn enters a cycle. Compute the length of this cycle.

For example, for the following array A:

```
A[0] = 2    A[1] = 3    A[2] = 1
A[3] = 1    A[4] = 3
```



```
A:  | 2 | 3 | 1 | 1 | 3 |
      0   1   2   3   4
```
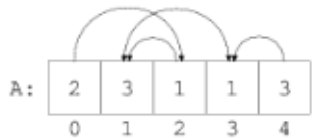
consecutive positions of the pawn are: 0, 2, 1, 3, 1, 3, 1, 3, ..., and the length of the cycle is 2.

Write a function:

```
public func solution(_ A : inout [Int]) -> Int
```

that, given a non-empty array A consisting of N integers in the range [0..N−1], returns the length of the cycle that the pawn eventually enters, as described above. For example, given the array shown above, the function should return 2.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..200,000];
- each element of array A is an integer within the range [0..N−1].

An array A consisting of N integers is given. A triplet (P, Q, R) is *triangular* if it is possible to build a triangle with sides of lengths A[P], A[Q] and A[R]. In other words, triplet (P, Q, R) is triangular if $0 \le P < Q < R < N$ and:

- A[P] + A[Q] > A[R],
- A[Q] + A[R] > A[P],
- A[R] + A[P] > A[Q].

For example, consider array A such that:

```
A[0] = 10    A[1] = 2    A[2] = 5
A[3] = 1     A[4] = 8    A[5] = 12
```

There are four triangular triplets that can be constructed from elements of this array, namely (0, 2, 4), (0, 2, 5), (0, 4, 5), and (2, 4, 5).

Write a function:

```
public func solution(_ A : inout [Int]) -> Int
```

that, given an array A consisting of N integers, returns the number of triangular triplets in this array.

For example, given array A such that:

```
A[0] = 10    A[1] = 2    A[2] = 5
A[3] = 1     A[4] = 8    A[5] = 12
```

the function should return 4, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..1,000];
- each element of array A is an integer within the range [1..1,000,000,000].

nab

CODILITY TASK EXAMPLES – TYPE SWIFT

Programming language: TypeScript ▼   **Spoken language:** English ▼

Build a like button component using Angular (v4). Export the like button component as "LikeButtonComponent" (`export class LikeButtonComponent`).

Requirements:

1. There should be a like button:

- The content of the like button should be in the following format: "Like | 100", where 100 is the total number of likes.
- It should have a "like-button" class.
- Wrap the number of likes in a span with a "likes-counter" class.
- The initial number of likes in the counter should be 100.

2. Users can add a like. By clicking the button:

- The number of likes should increase by one.
- Like button should have "liked" class in addition to the "like-button" class.

3. Users can undo their like by clicking again on the button:

- The counter should decrease by one.
- "liked" class should be removed.

Assessment/Tools:

- Use the animation below as a reference for your solution.
- Design/styling is not assessed and will not affect the score. You should focus only on implementing the requirements.
- The "Preview" tab will display your component. You can use it for testing purposes.

Like | 100

Programming language: TypeScript ▼

Implement a Strategy Design Pattern solution for a payment method in a shopping cart app using TypeScript. There are two payments methods that need to be implemented: PayPal and credit card.

Requirements

1. Your code should compile.
2. Change the *AbsPaymentMode* class to be an abstract class and to use the abstract method `pay` to implement the strategy class.
3. The *ShoppingCart* is already implemented and does not need to be modified. This class serves as a context to make use of the strategy class.
4. Implement the *pay* method in the *PaymentCreditCard* class, observing the following rule: To make a payment with a credit card, you have to add 2.1% tax to the total payment.
5. Implement the *pay* method in the *PaymentPayPal* class, observing the following rule: The PayPal service imposes a cost of $1.50, so you must add this cost to the total payment.
6. The *pay* method accepts a `number` argument and should return a `number` as its result.
7. The total value should be rounded to two decimal places.
8. Remember to assign your classes to `module.exports`, otherwise all tests will fail.
9. For any inheritance use `extends`

Assumptions

- You can assume that only valid positive numeric values will be provided as input.

Hints

- Your solution will be evaluated based on its correctness; performance and coding style will not be assessed.

Available packages/libraries

- TypeScript *v4.1.3*

Examples

The *pay* method should work as illustrated by the following examples:

When PayPal is used to pay, the expected values are:

- pay(50) === 51.5
- pay(2) === 3.5
- pay(8.328) === 9.83
- pay(4.57) === 6.07
- pay(2540.6) === 2542.1

When a credit card is used to pay, the expected values are:

- pay(50) === 51.05
- pay(2) === 2.04
- pay(8.328) === 8.50
- pay(4.57) === 4.67
- pay(2540.6) === 2593.95

nab

CODILITY TASK EXAMPLES – VISUAL BASIC

Programming language: Visual Basic ▾

A six-sided die is a small cube with a different number of pips on each face (side), ranging from 1 to 6. On any two opposite sides of the cube, the number of pips adds up to 7; that is, there are three pairs of opposite sides: 1 and 6, 2 and 5, and 3 and 4.

There are N dice lying on a table, each showing the pips on its top face. In one move, you can take one die and rotate it to an adjacent face. For example, you can rotate a die that shows 1 so that it shows 2, 3, 4 or 5. However, it cannot show 6 in a single move, because the faces with one pip and six pips visible are on opposite sides rather than adjacent.

You want to show the same number of pips on the top faces of all N dice. Given that each of the dice can be moved multiple times, count the minimum number of moves needed to get equal faces.

Write a function:

```
Private Function solution(A As Integer()) As Integer
```

that, given an array A consisting of N integers describing the number of pips (from 1 to 6) shown on each die's top face, returns the minimum number of moves necessary for each die to show the same number of pips.

For example, given:

- A = [1, 2, 3], the function should return 2, as you can pick the first two dice and rotate each of them in one move so that they all show three pips on the top face. Notice that you can also pick any other pair of dice in this case.

- A = [1, 1, 6], the function should also return 2. The only optimal answer is to rotate the last die so that it shows one pip. It is necessary to use two rotations to achieve this.

- A = [1, 6, 2, 3], the function should return 3. For instance, you can make all dice show 2: just rotate each die which is not showing 2 (and notice that for each die you can do this in one move).

Assume that:

- N is an integer within the range [1..100];
- each element of array A is an integer within the range [1..6].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Programming language: Visual Basic ▾   Spoken language: English ▾

You would like to set a password for an email account. However, there are two restrictions on the format of the password. It has to contain at least one uppercase character and it cannot contain any digits.

You are given a string S consisting of N alphanumerical characters. You would like to find the longest substring of S that is a valid password. A substring is defined as a contiguous segment of a string.

For example, given "a0Ba", the substrings that are valid passwords are "B" and "Ba". Note that "aBa" is not a substring and "a0B" is not a valid password.

Write a function:

```
Private Function solution(S As String) As Integer
```

that, given a non-empty string S consisting of N characters, returns the length of its longest substring that is a valid password. If there is no such substring, your function should return −1.

For example, given "a0Ba", your function should return 2, as explained above. Given "a0bb", your function should return −1, since there is no substring that satisfies the restrictions on the format of a valid password.

Assume that:

- N is an integer within the range [1..200];
- string S consists only of alphanumerical characters (a–z and/or A–Z and/or 0–9).

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

nab