

1) You are given an array prices where prices[i] is the price of a given stock on the ith day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

Example 1:

Input: prices = [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]

Output: 0

Explanation: In this case, no transactions are done and the max profit = 0...

2) Given a sequence of length ≤ 8 consisting of I and D, where I denotes the increasing sequence and D denotes the decreasing sequence, decode the sequence to construct a minimum number without repeated digits.

For example,

sequence	output
----------	--------

IIDDIDID	→ 125437698
----------	-------------

IDIDII	→ 1325467
--------	-----------

DDDD	→ 54321
------	---------

IIII	→ 12345
------	---------

3) Evaluate a postfix expression

Write code to evaluate a given postfix expression efficiently.

For example,

82/ will evaluate to 4 (8/2)

138*+ will evaluate to 25 (1+8*3)

545*+5/ will evaluate to 5 ((5+4*5)/5)

Assume that the postfix expression contains only single-digit numeric operands, without any whitespace.

4) Do the following problems without using multiplication or division operators

- a) Multiply a given integer with 15
- b) Multiply a given even integer with 7.5
- c) Calculate value of $15 \times n/16$

5) Find whether an array is subset of another array

Sample 1 : Input: arr1[] = {11, 1, 13, 21, 3, 7}, arr2[] = {11, 3, 7, 1}

Output: arr2[] is a subset of arr1[]

Sample 2 : Input: arr1[] = {1, 2, 3, 4, 5, 6}, arr2[] = {1, 2, 4}

Output: arr2[] is a subset of arr1[]

Sample 3 : Input: arr1[] = {10, 5, 2, 23, 19}, arr2[] = {19, 5, 3}

Output: arr2[] is not a subset of arr1[]

6) Given an array containing only 0s and 1s, find the largest subarray which contains equal no of 0s and 1s. The expected time complexity is $O(n)$.

Sample 1 : Input: arr[] = {1, 0, 1, 1, 1, 0, 0}

Output: {0, 1, 1, 1, 0, 0}

Explanation : from index 1 to 6 (Starting and Ending indexes of output subarray)

Sample 2 : Input: arr[] = {1, 1, 1, 1}

Output: No such subarray

Sample 3 : Input: arr[] = {0, 0, 1, 1, 0}

Output: {0, 0, 1, 1} or {0, 1, 1, 0}

Explanation : from index {0 to 3} Or {1 to 4}

13) Given an integer n, we need to find how many digits remove from the number to make it a perfect square.

Examples :

Input : 8314

Output: 81 2

Explanation: If we remove 3 and 4 number becomes 81 which is a perfect square.

Input : 57

Output : -1

9) Write a program to print all the LEADERS in the array. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader.

x

Example :

Input: {16, 17, 4, 3, 5, 2}

Output: 17, 5 and 2

7) Given a string S. The task is to print all unique permutations of the given string in lexicographically sorted order.

Example 1:

Input: ABC

Output: ABC ACB BAC BCA CAB CBA

Explanation:

Given string ABC has permutations in 6

forms as ABC, ACB, BAC, BCA, CAB and CBA .

Example 2:

Input: AB SG

Output:

ABGS AB SG AGBS AG SB ASBG AS GB BAGS

BASG BGAS BGSA BSAG BSGA GABS GASB

GBAS GBSA GSAB GSBA SABG SAGB SBAG

SBGA SGAB SGBA

Explanation:

Given string AB SG has 24 permutations.

8) Given two dates, find total number of days between them. The count of days must be calculated in $O(1)$ time and $O(1)$ auxiliary space.

Examples:

Input: $dt1 = \{10, 2, 2014\}$

$dt2 = \{10, 3, 2015\}$

Output: 393

$dt1$ represents "10-Feb-2014" and $dt2$ represents "10-Mar-2015"

The difference is $365 + 28$

Input: $dt1 = \{10, 2, 2000\}$

$dt2 = \{10, 3, 2000\}$

Output: 29

Note that 2000 is a leap year

Input: $dt1 = \{10, 2, 2000\}$

$dt2 = \{10, 2, 2000\}$

Output: 0

Both dates are same

Input: $dt1 = \{1, 2, 2000\};$

$dt2 = \{1, 2, 2004\};$

Output: 1461

Number of days is $365 \times 4 + 1$

10) Given a number, the task is that we divide number by 3. The input number may be large and it may not be possible to store even if we use long long int.

Sample 1 - Input : n = 769452

Output : Yes

Sample 2 - Input : n = 123456758933312

Output : No

Sample 3 - Input : n = 3635883959606670431112222

Output : Yes

11) Given an array health[] where health[i] is the health of the ith player in a game, any player can attack any other player in the game. The health of the player being attacked will be reduced by the amount of health the attacking player has. The task is to find the minimum possible health of the winning player. (PS- Always the player with lower health will attack the player with higher health)

Sample 1 - Input: health[] = {4, 6, 8}

Output: 2

4 attacks 6, health[] = {4, 2, 8}

2 attacks 4 twice, health[] = {0, 2, 8}

2 attacks 8 four times, health[] = {0, 2, 0}

Sample 2 - Input: health[] = {4, 1, 5, 3}

Output: 1

12) Given a rectangular sheet of length l and width w . we need to divide this sheet into square sheets such that the number of square sheets should be as minimum as possible.

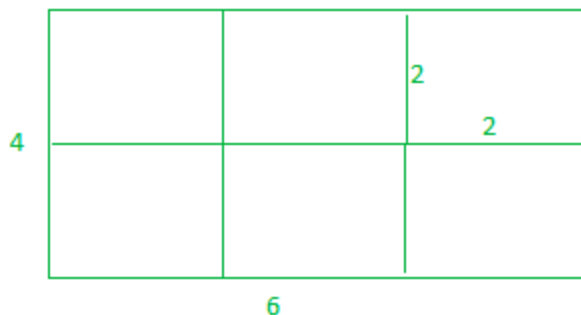
Examples:

Input : $l=4$ $w=6$

Output :6

We can form squares with side of 1 unit, But the number of squares will be 24, this is not minimum. If we make square with side of 2, then we have 6 squares. and this is our required answer.

And also we can't make square with side 3, if we select 3 as square side, then whole sheet can't be converted into squares of equal length.



Square having
side 2.
Number of
Square=6

Input : $l=3$ $w=5$

Output :15

13) Given an integer n, we need to find how many digits remove from the number to make it a perfect square.

Examples :

Input : 8314

Output: 81 2

Explanation: If we remove 3 and 4 number becomes 81 which is a perfect square.

Input : 57

Output : -1

14) Given a string s, print all possible subsequences of the given string in an iterative manner. We have already discussed Recursive method to print all subsequences of a string.

Examples:

Input : abc

Output : a, b, c, ab, ac, bc, abc

Input : aab

Output : a, b, aa, ab, aab

15) You are given a string that represent an expression of digits and operands. E.g. $1+2*3$, $1-2+4$. You need to evaluate the string or the expression. NO BODMAS is followed. If the expression is of incorrect syntax return -1.

Test cases:

a) $1+2*3$ will be evaluated to 9.

b) $4-2+6*3$ will be evaluated to 24.

c) $1++2$ will be evaluated to -1(INVALID).

Also, in the string spaces can occur. For that case we need to ignore the spaces. Like :- $1*2-1$ is equals to 1.

16) Find the seating position of passengers on a flight. The seating arrangement of a flight will be as shown in the picture.

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30
36	35	34	33	32	31
37	38	39	40	41	42
48	47	46	45	44	43
49	50	51	52	53	54
60	59	58	57	56	55
61	62	63	64	65	66
72	71	70	69	68	67
73	74	75	76	77	78
84	83	82	81	80	79
85	86	87	88	89	90
96	95	94	93	92	91
97	98	99	100	101	102
108	107	106	105	104	103

There are three types of seat positions

1.Window Seat (WS) 2. Middle Seat (MS) 3. Aisle Seat (AS)

Given a number print the corresponding seat position.

Sample Input 1: 63

Sample Output 1: AS

Sample Input 2: 77

Sample Output 2: MS

17) Write a program to in-place move all zeros to the end in the given integer array.
Without using extra array

Sample Input 1:

8

{1, 2, 0, 4, 3, 0, 5, 0}

Sample Output 1: {1,2,4,3,5,0,0,0}

Sample Input 2:

8

{9,2,0,4,3,0,2,0}

Sample Output 2: {9,2,4,3,2,0,0,0}