CODEKNACK 2018

January – February

AIM

The aim of the contest was to imbibe a basic development of logic in FY B. Tech students, which might have helped them understand their own flaws in coding. We consider self-help to be the best kind of help, hence we chose and selected a variety of easy and tough questions for CODEKNACK 2018.

WHY WE CHOSE HackerRank

Flexibility in solving questions as per user's wish, along with online compiling in more than 30 coding languages, was something which we wanted to be implemented in this contest.

SCORING

The scoring was done as was mentioned on the contest sign-up link:

- Each challenge has a pre-determined score
- A participant's score depends on the number of test cases his/her submission successfully passes.
- In case of multiple submissions for the same challenge, the highest score obtained was reflected.
- Participants are ranked by score. If two or more participants achieve the same score, the tie is broken by the total time taken to submit the last solution resulting in a higher score.

COST

After careful consideration, we decided to keep NO registration and participation fees, since we believe in a NO-FEE-FOR-CODE policy. Although this prevented us from providing better gifts for the winners, we want coders to feel satisfied purely by just solving each challenge and not feeling restricted due to money.

C H A L L E N G E 1

LINK SHORTENER

(20 points)

You are in charge of a newly created company, called KnackShort, aimed at mainly link shortening. Unfortunately, tomorrow is KnackShort's opening day, and your head programmer has taken ill.
Write a program to input a website name, shorten it, and add it to the end of: "*www.ks.com/...*"
Rules for shortening:

- Remove www. from input website
- Output has no vowels
- Top level of domain (i.e .com, .org, **.edu.in** etc) needs to remain in output

Input Format

```
www.kalti.com
```

Constraints

1 ≤ L ≤ 200

where L is the length of the website.

Output Format

```
www.ks.com/klt.com
```

Sample Input 0

```
www.kalti.com
```

Sample Output 0

```
www.ks.com/klt.com
```

Explanation 0

kalti becomes klt as per rules mentioned.

Agenda:

- Being a basic string operations program, this problem judges your ability to handle string variables, including making substrings, and considering or ignoring characters.

CHALLENGE 2

ROTATE AN ARRAY

(20 points)

Given an array of n integers, perform rotation of elements (1st to last, last to second last, and so on). The first line of input is the number of rotations and second line is the array. Consider Array of minimum 5 elements and maximum 10 elements.

Input Format

4
1 2 3 4 5

Constraints

Each element can be considered as a standard 32-bit integer.

Output Format

The array after 4 rotations:
5 1 2 3 4

Sample Input 0

4
1 2 3 4 5

Sample Output 0

The array after 4 rotations:
5 1 2 3 4

Tips:

- Being an array operations program, this problem judges your ability to handle index positions, and replacing of numbers.

CHALLENGE 3

ARRAY'S SWAYING ARRANGEMENT (30 points)

Write a program to input a list of integers in an array and arrange them in a **swaying fashion** such that:

- The smallest element out of the list of odd number of integers, must come at the center position of array.
- The number in the ascending order next to the minimum, goes to the right, the next higher number goes to the left of minimum number and so on.

Note: Consider only 5 elements in input and output array.

Input Format

1 2 3 4 5

Constraints

- Each element can be considered as a standard 32-bit integer.
- Input elements are not necessarily in ascending order; hence you might want to sort the input array.

Output Format

The array, after being arranged in swaying fashion:
5 3 1 2 4

Sample Input 0

1 2 3 4 5

Sample Output 0

The array, after being arranged in swaying fashion:
5 3 1 2 4

Explanation 0

As 1 is the least number, it is brought to the middle, 2 to the right of it, and 3 to the left of it.
Further, 4 is then brought to the right of 2, and 5 to the left of 3 as stated in question.

Tips:

- Being an array operations program, this problem judges your ability to handle index positions, and replacing of numbers.
- Some might prefer to use odd and even index positions, however solutions can either be naïve (using basic logic) or efficient (using data structures or similar concepts – language specific) depending on how you want to implement it.

CHALLENGE 4

STRING ARRANGEMENTS (30 points)

Write a program to accept a word and output all the letters in alphabetical order. If the same letter repeated twice, it should be printed twice.
Inputs and outputs in lowercase only.

Input Format

carrot

Constraints

The inputs should be single words(No spaces).

Output Format

acorrt

Sample Input 0

carrot

Sample Output 0

acorrt

Tips:

- This is a simple program built on the concept of sorting characters alphabetically.
- The judgement parameter here is not exactly the final output, since that is achieved quite easily, but the precise means of getting the output. For example, naïve solutions of Bubble or Selection sort may not fetch you as many points as you would obtain using internal sorting implementations or lists/hash tables,

CHALLENGE 5

MARATHON GIFTS

(50 points)

Back in the year 1995, Pepsipi was the regional marathon runner for Codbay.
Pepsipi's coach was concerned about her running, which somehow degraded as days passed. He found out a way to get her to win.
He recorded her timing daily in milliseconds (1/1000th of a second), and devised a scheme whereby each time she made a win with timing lower than the previous day's timing, by at least a certain number of milliseconds, he would reward her with a bronze coin as a gift.

Given, a list of daily timings for Pepsipi's runs, **determine how many coins Pepsipi would have received.**

Input Format

The first line will contain two integers, **a** & **k**, where **a** is the number of days, and **k** is the desired improvement (in milliseconds). Whenever Pepsipi's timing reduces by at least milliseconds over the previous day's timing, she will get a bronze coin from the coach.
The first line is followed by lines, where each lines contains a single integer, which is Pepsipi's daily timings in milliseconds. The timings are in chronological order.

```
6 100
59420
59410
59310
59290
59470
59350
```

Constraints

$3 \leq a \leq 1{,}000$

$0 < k \leq 100{,}000$

$0 < t \leq 100{,}000$

Output Format

```
2
```

Tips:

- This is a simple program based on whether the coder can map the given problem to a similar data representation
- The basic program is quite simple, but the various constraints and the type of conditions imposed are what the coder needs to map.

CHALLENGE 6

WORD CLOCK

(50 points)

If we have a time in numbers, (thanks to an electronic clock signal), it can be transformed/translated (by code) into words. For example :

- 7 : 00 —— seven o'clock
- 7 : 10 —— ten minutes past seven
- 7 : 15 —— quarter past seven
- 7 : 30 —— half past seven
- 7 : 40 —— twenty minutes to eight
- 7 : 45 —— quarter to eight
- 7 : 47 —— thirteen minutes to eight
- 7 : 60 —— Incorrect Input

Write a program which **inputs two integers**.

- The first integer between 1 and 12 (both inclusive, to represent hours) and
- second integer between 0 and 59 (both inclusive, to represent minutes)
- and then prints out the time they represent, in words, as stated in the examples.

Input Format

3 0

Constraints

- Assume basic 32-bit integers.
- **Capitalise the first letter** of each output.

Output Format

3 : 00 Three o' clock

Sample Input 0

3 0

Sample Output 0

3 : 00 Three o' clock

Explanation 0

As 0 minutes past 3, it should output o'clock.

Tips:

- This is a simple program built on the concept of conditional statements of the minute parameter

Sample Input 1

7 29

Sample Output 1

7 : 29 Twenty nine minutes past Seven

Explanation 1

As 29 minutes elapsed, within conditions of quarter past and half past, output the corresponding minutes.

Sample Input 2

6 34

Sample Output 2

6 : 34 Twenty six minutes to Seven

Sample Input 3

14 60

Sample Output 3

Incorrect Input

Explanation 3

As minutes are within 0 to 59.

CHALLENGE 7

MAXIMUM INVESTORS' TAX (100 points)

A fictional city called Codbay consists of buildings, each of equal length and breadth.

Each building has a certain number of permanent investors, represented as a positive integer. Potential investors are represented by negative integers. Note that, each building can have either permanent or potential investors, not both.

The ruling party, which has recently come into power, decides to pass a new tax, called "M.I.T" or Maximum Investors Tax, which states that:
" *The contiguous buildings/areas with the largest sum of the numbers of permanent or potential investors , shall be applicable for deposition of M.I.T within 30 days of procurement of profits.*"
.....

You have been hired as a coder for Codbay's ruling party to determine the contiguous buildings which are liable for paying M.I.T.

- Input two integers first, indicating rows and columns respectively, to ensure changes in future city plans.
- Take the number of permanent or potential investors as integer values in a two-dimensional array.
- Output the buildings(represented by investors' number) which need to pay M.I.T (i.e, have the largest sum contiguously), in the pattern as inputted.

Input Format

```
6  5
0  -2  -7  0  -1
9  2  -6  2  0
-4  1  -4  1  0
-1  8  0  -2  1
-10 1  1  -5  6
-15 -1  1  5  -4

//all integers are tab-separated
```

Constraints

- Consider 32-bit Integer numbers
- If all numbers are negative, the answer would be the buildings with the total sum closest to 0.

Output Format

```
9  2
-4  1
-1  8
```

Tips:

- This is a complex program, a typical example of dynamic programming.
- The basic task is to find the submatrix with the greatest sum of a given matrix
- Although a number of solutions exist, with improvements using language-specific features, the question is framed on Kadane's algorithm. You can base your solution on the video below:

https://www.youtube.com/watch?v=yCQN096CwWM

Sample Input 0

```
6  5
0  -2  -7  0  -1
9  2  -6  2  0
-4  1  -4  1  0
-1  8  0  -2  1
-10  1  1  -5  6
-15  -1  1  5  -4
```

Sample Output 0

```
9  2
-4  1
-1  8
```

Explanation 0

The output buildings are the ones with the largest contiguous sum of investors.