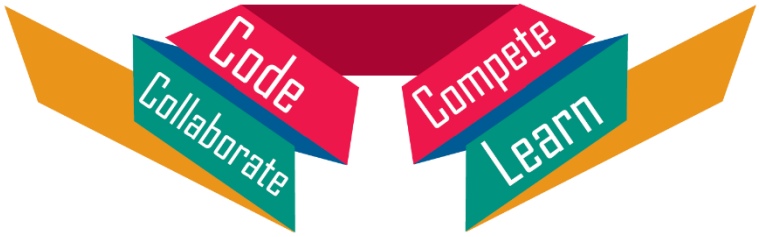
2022 NORTHERN INDIANA HIGH SCHOOL CODING COMPETITION

April 23, 2022



COMPUTER SCIENCE AND INFORMATICS
COLLEGE OF LIBERAL ARTS AND SCIENCES
INDIANA UNIVERSITY SOUTH BEND

Round One

Problem 1. **Julian Day Number**

The Julian Day Number is the number of days elapsed since Monday, January 1, 4713 BC. It is used primarily by astronomers, and in software for easily calculating elapsed days between two events.

Given a date (day, month, year), here is the formula to calculate Julian Day Number (*JDN*).

$$a = \frac{14 - month}{12}$$

$$y = year + 4800 - a$$

$$m = month + 12 * a - 3$$

$$JDN = day + \frac{153 * m + 2}{5} + 365 * y + \frac{y}{4} - \frac{y}{100} + \frac{y}{400} - 32045$$

Note: In above equations, all divisions are **integer divisions**, fractional parts are ignored (discarded).

For example, today's date is **April 23, 2022** (day is 23, month is 4, and year is 2022). Using the formula, we can determine today's Julian Day Number (JDN) is **2,459,693**.

Write a program using the Julian Day Number formula to determine how old you are today (how many days since you have been born).

Input of the program has one line representing your birth date in the format of **mm dd yyyy** separated by a white space, where **mm** is month, **dd** is day, and **yyyy** is year.

Notes: (1) The input date should not be later than today (April 23, 2022); (2) The input year should be greater than 1000; (3) You are not allowed to use any built-in Date or Calendar functions to solve this problem.

Sample input (red color) and output (blue color):

Input: 04 23 2022
Output: 0

Input: 04 22 2022
Output: 1

Input: 03 23 2022
Output: 31

Input: 04 23 2021
Output: 365

Problem 2. Credit Card Number

Credit card numbers are generated with some rules to ensure that one cannot come with a random sequence of 16 digits and guess a valid credit card number. One such rule would require that the **sum of digits** of the credit card **modulo 29 is equal to 5**. Actual credit cards use more sophisticated rules, but for this problem, we will assume that this simple rule applies. As a reminder, n mod 29 is equal to the remainder of the integer division of n by 29.

Write a program to validate a credit card number using the rule described above. The input line contains a 16-digit credit card number (no spaces between digits). The output will be either "valid" or "not valid" reflecting the result of modulo operation.

Sample input (red color) and output (blue color):

Input: 1000000000000004
Output: valid

Input: 1111111111111111
Output: not valid

Input: 6973498620404957
Output: not valid

Input: 8752738974903866
Output: valid

Problem 3. Text Editing

Given a string containing **many consecutive white spaces**, write a program to trim all while spaces so that all words should contain only a single white space between them. The solution should handle trailing and leading white spaces and also remove preceding white spaces before common punctuations, including period, comma, and question mark.

Input of your program is a string of sentences and output of your program is the edited sentences satisfying the previous described requirement.

Notes: (1) Input string will not contain tab keys and other punctuation marks other than **period**, **comma**, and **question mark**. (2) Input string has at most 300 characters.

Sample input (red color) and output (blue color):

Input:
Hello Friends . Welcome to IU South Bend .
Output:
Hello Friends. Welcome to IU South Bend.

Input:
How are you today ?
Output:
How are you today?

Input:
Monday is blue, Tuesday is grey , Wednesday is unknown.
Output:
Monday is blue, Tuesday is grey, Wednesday is unknown.

Input:
Welcome our friends from Great Lake State . Do you like Michiana ?
Output:
Welcome our friends from Great Lake State. Do you like Michiana?

Problem 4. Vote Count

Given a list of votes for candidates to the position of President of St. Joseph County, IN, write a program to count the number of votes that each candidate gets and output the winner. The candidates are coded by a number going from 1 to n, and the votes mark which candidate was chosen by a voter. A voter can only cast a vote for one candidate.

The input of the program has two lines. The first line has two positive integer numbers (**c** and **v** separated by a white space) less than or equal to 20: **c** is the number of candidates and **v** is the total number of votes cast. The second line contains the list of votes cast by voters, which are separated by white spaces.

The output of the program has two lines. The first line should be the count of votes of each candidate in order, from 1 to n. The second line shows the winner (or winners). There should be white spaces between output numbers.

Sample input (red color) and output (blue color):

Input:
6 15
2 6 3 1 2 5 3 2 1 5 2 3 1 6 2

Explanation:
First line: 6→ there are 6 candidates (1, 2, 3, 4, 5, 6); 15→ 15 votes cast
*Second line: 15 numbers → 15 votes; each number **n** represents 1 vote received by candidate **n***

Output:
3 5 3 0 2 2
2

Explanation:
First line: 6 numbers→ the count of votes received by the 6 candidates (1, 2, 3, 4, 5, 6) in that order.
Second line: the winner candidate 2, who received the highest (5) votes.

Input:
5 17
1 3 2 2 4 1 2 2 1 1 4 4 4 2 1 3 4

Output:
5 5 2 5 0
1 2 4

Explanation:
Second line: the winners are candidates 1, 2, 4, who received the same highest (5) votes.

Problem 5. Triangles

Write a program to determine how many different triangles can be formed by using any three numbers from a given set (size is between 3 and 20 inclusive) of different positive numbers representing the length of sides of triangles.

Hint: Given three sides **a**, **b**, **c**, they can form a triangle if and only if the sum of any two sides is greater than the third side.

$$\begin{cases} a + b > c \\ a + c > b \\ b + c > a \end{cases}$$

For example, 11, 12, 13 can form a triangle while 11, 12, 40 cannot form a triangle.

The input of your program has two lines. The first line is a number representing the size of the set (number of sides). The second line contains the positive integer numbers with **different** values, separated by white spaces representing length of sides.

The output of your program is the number of different triangles that can be formed.

Sample input (red color) and output (blue color):

input:
4
1 4 3 2

output: 1

Explanation: only one triangle can be formed with sides (2, 3, 4).

input:
5
10 12 1 13 14

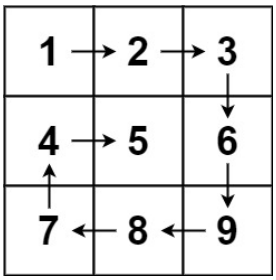
output: 4

Explanation: Four triangles can be formed: (10, 12, 13) (10, 12, 14) (10, 13, 14) (12, 13, 14)

Problem 6. 2D-Array Traversal

Given a 2D ($m \times n$) integer array, where m (number of rows) and n (number of columns) are greater than or equal to 1, write a program to print (display on screen) the array content in **clockwise spiral form** starting from the upper left corner.

For example, given this array,



Your program should display: 1 2 3 6 9 8 7 4 5

Input of your program has several lines. The first line has two numbers **m** and **n** separated by a white space, representing dimensions of the array (**m** rows and **n** columns). The rest of the lines are the contents of the array: each line contains one row and adjacent numbers on the same row are separated by a white space.

Output of your program should contain all the numbers (separated by a white space) in the array in spiral sequence (all in one line).

Sample input (red color) and output (blue color):

Input:
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

Output:
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Input:
3 6
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18

Output:
1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11