

C++ CODING WORKSHOP

Editorial

2022 C++ Coding Workshop Mini Contest

TC22A0 - Gonna Give You Up

TC22A0 - Gonna Give You Up

You know the rules, and you know how to do.

TC22A1 - Letter to the Editor

Problem Statement

Given an alphabetical letter M and a passage.
Find how many times the M appeared in the passage.

Subtask 1

No. of lines of the letter = 1
 $1 \leq \text{Length of the letter} \leq 30$

The letter contains no spaces

The letter contains no spaces $\rightarrow$ no need to worry anything
When inputting the string, just declare a string and use `cin`.

Subtask 1

The easiest way to solve this question:

Linear Search

Using for loop:

```
for (loop from 0 to size of the string){  
    if (the current letter == M)  
        count of letter + 1
```

Subtask 1 Done!

Score: 7

Full Solution

However, when there is a space, or the passage contains more than 1 line, the code using only one `cin` is not working.

Example:

Input a
 an apple

Output 1

But the correct answer is 2!

Full Solution

Why it is wrong?

```
[an] apple
```

The variable is only being input once, so the string will constantly store the word `an` only. How can we avoid this?

Using `getline(cin, the_string)` → `[an apple]` (get input of whole line)

Yet, it is still not working. Why?

Full Solution

After you have `cin` the character from the first line, the input pointer still remains on the first line.

```
a *[] (getting an empty line)
```

```
an apple
```

You have to use `getline` twice to guarantee that the correct line would be pointed.

But it is still not working!!!

Full Solution

Looking back to the input...

“The following unknown number of lines are the contents of the letter, which may contain lower-case English letters, numbers, punctuation marks, symbols, and spaces.”

The following unknown number of lines ← this is the problem

Full Solution

There may be more than 1 line of the input.

Example:

```
a  
an apple  
an apple
```

The situation if you used `getline`:

```
[a]  
[an apple]  
an apple
```

You can only check one line but not all.

Full Solution

How can we solve this problem?

New technique: `while(cin >> S)`

- Combine `while` and `cin`
- Input all stuff into `S` until there are nothing in the input left

```
while (cin >> S){  
    ...  
}
```

Full Solution

How to implement?

Do it by yourself! :)

Score: 10

TC22A2 - Diskalmer

Problem Statement

Given N English vocabularies, if the number of distinct English vocabularies is greater or equal to M , output **Non-tutor**, otherwise, output **Tutor**.

Subtask 1 & 2

Make a vector (an array is also OK) .

For each string, check if the number is in the vector or not.

If no, add it into the vector.

Linear time search, N queries.

Time complexity: $O(N^2)$

Score: 17

Full Solution

Use a data structure called `set`.

It can achieve $\lg N$ time for finding a string in the set or not.

Time complexity: $O(n \lg n)$

Score: 20

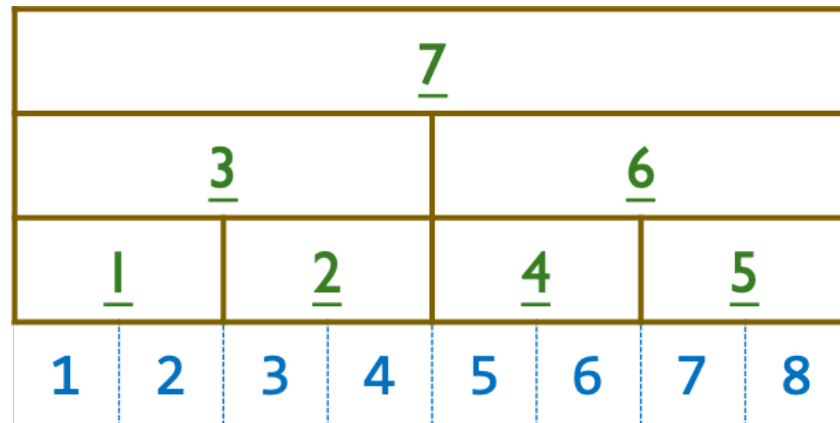
TC22A3 - Bob the Builder

Problem Explanation

Think about the top layer, to build this layer, you have to build the left part and right part first to build the top layer.

Using 3 as example, when you need to build 1 8, you have to build 1 4 and 5 8 before that.

Using recursive idea, to build 1 4, 1 2 and 3 4 is needed beforehand.



Problem Explanation

We know that each time to build a layer, the layer below it has to be done. Therefore, we can transform this idea into a series of action:

`solve(int left_bound, int right_bound):`

1. Solve the left part at the lower layer
2. Solve the right part at the lower layer
3. Build the current layer

At the beginning, we call `solve(1,  $2^N$ )`, as the top layer have 1 as left bound and 2^N as right bound.

But we need to add a base case: when `(l == r)`, such as `l = 1, r = 1`, then we can stop this branch as it already go to the most bottom layer.

TC22A4 - Two Stone One Bird

Problem Statement

Given S stones and B birds, output the number of “two birds one stone” and “two stone one birds” situation happen.

If the situation is impossible, output `Impossible`.

Subtask 1

$$0 \leq S, B \leq 2$$

Hard-code.

There are total of 9 cases.

Score: 5

B \ S	0	1	2
0	0 0	N/A	N/A
1	N/A	N/A	0 1
2	N/A	1 0	N/A

Subtask 2

$$0 \leq S, B \leq 100$$

There is no “one stone two birds” situation happened

It is guaranteed that the data given is “valid”

Just output θ B.

Score: 5 (Cumulative: 10)

Subtask 3

$$0 \leq S, B \leq 100$$

There is no “two stones one bird” situation happened

It is guaranteed that the data given is “valid”

Same idea as Subtask 2.

Just output $S \setminus \emptyset$.

Score: 5 (Cumulative: 15)

Subtask 4

$$0 \leq S, B \leq 100$$

It is guaranteed that the data given is “valid”

Constraints are small!

We can use some loops (for loop, or while loop, up to you) to test all combinations of output between 0 0 to 50 50.

Score: 5 (Cumulative: 20)

Subtask 5

It is guaranteed that the data given is “valid”

Oh... The loops will grant us TLE...

See the hint? Let us use some Maths.
In specific, let us use simultaneous equation.

HINT

Use mathematical knowledge for a full mark solution.

Subtask 5

Let

“one stone two birds” $\rightarrow x$

“two stones one bird” $\rightarrow y$

We can evaluate these two equations:

$$\begin{cases} x + 2y = S \\ 2x + y = B \end{cases}$$

Subtask 5

$$\begin{cases} x + 2y = S \\ 2x + y = B \end{cases}$$

The problem is simple now!

By solving the equations...

$$\begin{cases} x = \frac{2B - S}{3} \\ y = \frac{2S - B}{3} \end{cases}$$

Subtask 5

$$\begin{cases} x = \frac{2B - S}{3} \\ y = \frac{2S - B}{3} \end{cases}$$

Done!

Time complexity: $O(1)$

Score: 10 (Cumulative: 30)

Full Solution

We have to consider when the data given is “invalid”.

When will it be invalid?

$$\begin{cases} x = \frac{2B - S}{3} \\ y = \frac{2S - B}{3} \end{cases}$$

When $2B - S$ or $2S - B$ is not divisible by 3.

Full Solution

Add an `if` statement:

If $2B - S$ or $2S - B \bmod 3$ not equals to 0, or $2B - S < 0$, or $2S - B < 0$,
output `Impossible`.

Done!

Score: 35

Thank You!