Codeforces Beta Round #5

Problema B: *Center Alignment*

Prof. Edson Alves – UnB/FGA

Codeforces Beta Round #5 – Problem B: Center Alignment

Problema

Almost every text editor has a built-in function of center text alignment. The developers of the popular in Berland text editor «Textpad» decided to introduce this functionality into the fourth release of the product.

You are to implement the alignment in the shortest possible time. Good luck!

Entrada e saída

Input

The input file consists of one or more lines, each of the lines contains Latin letters, digits and/or spaces. The lines cannot start or end with a space. It is guaranteed that at least one of the lines has positive length. The length of each line and the total amount of the lines do not exceed 1000.

Output

Format the given text, aligning it center. Frame the whole text with characters «*» of the minimum size. If a line cannot be aligned perfectly (for example, the line has even length, while the width of the block is uneven), you should place such lines rounding down the distance to the left or to the right edge and bringing them closer left or right alternatively (you should start with bringing left). Study the sample tests carefully to understand the output format better.

Exemplo de entradas e saídas

Sample Input

```
This  is

Codeforces
Beta
Round
5
```

Sample Output

```
************
* This  is *
*          *
*Codeforces*
*   Beta   *
*  Round   *
*     5    *
************
```

Solução com complexidade $O(n)$

- Primeiramente, é preciso determinar o tamanho da maior linha M, o qual irá determinar a largura do quadro
- Inicialmente, deve ser impressa uma linha com $M + 2$ caracteres '*'
- Para cada linha L, é preciso confrontar seu tamanho com M
- Se for menor, a diferença deve ser dividida igualmente entre os lados esquerdo e direito
- No caso de uma diferença ímpar, o caractere restante deve ser distribuído alternadamente, inicialmente à direita
- Uma variável auxiliar *padding* pode ser usada para manter esta alternância
- Finalmente, deve ser impressa uma nova linha com $M + 2$ caracteres '*'

Solução AC

```cpp
#include <bits/stdc++.h>

using namespace std;

string solve(const vector<string>& lines)
{
    size_t max_size = 0, padding = 0;

    for (const auto& line : lines)
        max_size = max(max_size, line.size());

    ostringstream os;

    for (size_t i = 0; i < max_size + 2; ++i)
        os << '*' << (i == max_size + 1 ? "\n" : "");

    for (const auto& line : lines)
    {
        auto size = line.size();
        auto diff = max_size - size;
```

Solução AC

```
22          int right = diff / 2;
23
24          if (diff & 1) {
25              left += padding;
26              right += 1 - padding;
27              padding = 1 - padding;
28          }
29
30          os << '*';
31
32          while (left--)
33              os << ' ';
34
35          os << line;
36
37          while (right--)
38              os << ' ';
39
40          os << "*\n";
41      }
```

Solução AC

```
43      for (size_t i = 0; i < max_size + 2; ++i)
44          os << '*' << (i == max_size + 1 ? "\n" : "");
45
46      return os.str();
47  }
48
49  int main()
50  {
51      vector<string> lines;
52      string line;
53
54      while (getline(cin, line))
55          lines.emplace_back(line);
56
57      auto ans = solve(lines);
58
59      cout << ans;
60
61      return 0;
62  }
```