OJ 10935

Throwing Cards Away I

Prof. Edson Alves – UnB/FGA

Given is an ordered deck of n cards numbered 1 to n with card 1 at the top and card n at the bottom. The following operation is performed as long as there are at least two cards in the deck:

Throw away the top card and move the card that is now on the top of the deck to the bottom of the deck.

Your task is to find the sequence of discarded cards and the last, remaining card.

Input

Each line of input (except the last) contains a number $n \leq 50$. The last line contains '0' and this line should not be processed.

Output

For each number from the input produce two lines of output. The first line presents the sequence of discarded cards, the second line reports the last remaining card. No line will have leading or trailing spaces. See the sample for the expected format.

Exemplo de entradas e saídas

Sample Input

```
7
19
10
6
0
```

Sample Output

```
Discarded cards: 1, 3, 5, 7, 4, 2
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 8, 12, 16, 2, 10, 18, 14
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 2, 6, 10, 8
Remaining card: 4
Discarded cards: 1, 3, 5, 2, 6
Remaining card: 4
```

Solução com complexidade $O(N)$

- O processo descrito no problema pode ser simulado através do uso de uma fila
- Os elementos removidos podem ser armazenados ou em uma outra fila ou em um vetor
- Como, a cada ciclo, o tamanho da fila diminui em uma unidade, o algoritmo tem complexidade $O(N)$
- A saída deve ser formatada com cuidado: no caso $n = 1$ não há elementos a serem removidos

Solução com complexidade $O(N)$

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

using namespace std;

vector<int> solve(int n)
{
    queue<int> q;

    for (int i = 1; i <= n; ++i)
        q.push(i);

    vector<int> ans;

    while (q.size() > 1)
    {
        auto x = q.front();
        q.pop();

        ans.push_back(x);
```

```
21          auto y = q.front();
22          q.pop();
23
24          q.push(y);
25      }
26
27      ans.push_back(q.front());
28
29      return ans;
30 }
31
32 int main()
33 {
34      int n;
35
36      while (cin >> n, n)
37      {
38          auto order = solve(n);
39
40          cout << "Discarded cards:";
```

```
42          for (int i = 0; i < n - 1; ++i)
43              cout << (i ? ", " : " ") << order[i];
44
45          cout << "\nRemaining card: " << order.back() << '\n';
46      }
47
48      return 0;
49 }
```