

Codeforces Round #292 (Div. 2)

Problema C: *Drazil and Factorial*

Prof. Edson Alves – UnB/FGA

Drazil is playing a math game with Varda.

Let's define $F(x)$ for positive integer x as a product of factorials of its digits. For example, $F(135) = 1! \times 3! \times 5! = 720$.

First, they choose a decimal number a consisting of n digits that contains at least one digit larger than 1. This number may possibly start with leading zeroes. Then they should find maximum positive number x satisfying following two conditions:

- 1. x doesn't contain neither digit 0 nor digit 1.*
- 2. $F(x) = F(a)$.*

Help friends find such number.

Input

The first line contains an integer n ($1 \leq n \leq 15$) – the number of digits in a .

The second line contains n digits of a . There is at least one digit in a that is larger than 1. Number a may possibly contain leading zeroes.

Output

Output a maximum possible integer satisfying the conditions above. There should be no zeroes and ones in this number decimal representation.

Exemplos de entrada e saída

Entrada

4

1 2 3 4

3

5 5 5

Saída

3 3 2 2 2

5 5 5

- Para determinar o valor de x é preciso, inicialmente, determinar a fatoração prima de $F(a)$
- Como $F(a)$ é o produto do fatorial de cada dígito de a , esta fatoração conterá, no máximo, 4 primos distintos: 2, 3, 5 e 7
- Esta fatoração será composta pelo produto das fatorações de cada dígito de a
- Uma vez que há apenas 10 dígitos decimais e alguns deles podem se repetir em a , podemos usar um histograma para evitar o cálculo de uma mesma fatoração repetidas vezes

- Observe que o menor fatorial que contém o primo p em sua fatora  o   $p!$
- Como desejamos o maior x poss  vel, podemos escolher, gulosamente, o maior dentre os fatoriais $2!$, $3!$, $5!$ e $7!$ que ainda pode ser formado com os fatores dispon  veis
- A cada fatorial escolhido   preciso atualizar a lista dos fatores dispon  veis
- Eventualmente o resultado pode exceder os limites de um **long long**, ent  o utilize uma string para armazenar o resultado, evitando assim o *overflow*

```
18 map<int, int> fact_factorization(int n)
19 {
20     map<int, int> fs;
21
22     for (auto p : { 2, 3, 5, 7 })
23         fs[p] = E(n, p);
24
25     return fs;
26 }
```

```
28 vector<int> histogram(long long n)
29 {
30     vector<int> hs(10, 0);
31
32     while (n)
33     {
34         ++hs[n % 10];
35         n /= 10;
36     }
37
38     return hs;
39 }
```


Solução $O(n \log n)$

```
41 string solve(long long a)
42 {
43     auto hs = histogram(a);
44     map<int, int> fs;
45
46     for (int i = 2; i <= 9; ++i)
47         for (auto [p, k] : fact_factorization(i))
48             fs[p] += k*hs[i];
49
50     string ans;
51
52     for (auto p : { 7, 5, 3, 2 })
53     {
54         auto qs = fact_factorization(p);
55         auto n = fs[p];
56     }
```

Solução $O(n \log n)$

```
57     for (auto q : { 2, 3, 5, 7 })
58         if (q <= p)
59             n = min(n, fs[q] / qs[q]);
60
61     for (int i = 0; i < n; ++i)
62         ans.push_back((char) ('0' + p));
63
64     for (auto q : { 2, 3, 5, 7 })
65         fs[q] -= (n*qs[q]);
66 }
67
68 return ans;
69 }
```