逻辑题：1096 Consecutive Factors (20分)

PAT甲级：1096 Consecutive Factors (20分)

Among all the factors of a positive integer N, there may exist several consecutive numbers. For example, 630 can be factored as 3×5×6×7, where 5, 6, and 7 are the three consecutive numbers. Now given any positive N, you are supposed to find the maximum number of consecutive factors, and list the smallest sequence of the consecutive factors.

Input Specification:

Each input file contains one test case, which gives the integer N (1<N<231).

Output Specification:

For each test case, print in the first line the maximum number of consecutive factors. Then in the second line, print the smallest sequence of the consecutive factors in the format `factor[1]*factor[2]*...*factor[k]`, where the factors are listed in increasing order, and 1 is NOT included.

Sample Input:

```
630
```

Sample Output:

```
3
5*6*7
```

题目大意：

一个正整数N的因子中可能存在若干连续的数字。例如630可以分解为3*5*6*7，其中5、6、7就是3个连续的数字。给定任一正整数N，要求编写程序求出最长连续因子的个数，并输出最小的连续因子序列~

思路：

1. 因数的最大值不超过$\sqrt{number}+1$
2. 于是我们要做的就是从2开始构造一个最长的连乘。
3. 暴力历遍连乘，记录下个数最多的一组连乘。

代码参考：

```
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;
```

```
long int number = 0, temp = 0;
int main()
{
    int factor = 2, inf = 0, len = 0, first = 0;
    cin >> number;
    inf = sqrt(number)+1;
    vector<int> ans;
    for(factor=2;factor<inf;factor++)
    {
        int j, temp = 1;
        for(j=factor;j<=inf;j++)
        {
            temp *= j;
            if (number%temp != 0) break;
        }
        if(j-factor>len)
        {
            len = j - factor;
            first = factor;
        }
    }
    if (first == 0) cout << 1 << endl << number;
    else
    {
        cout << len << endl;
        for(int i=0;i<len;i++)
        {
            cout << first + i;
            if (i != len - 1) cout << "*";
        }
    }
    return 0;
}
```