**Анализ на решението на задача  
РАЗУЗНАВАЧ**

Трябва да намерим **най-малкото число**, чието произведение на цифрите е равно на ***K.*** По-точно, отговорът трябва да съдържа **възможно най-малък брой цифри**. Също така отбелязваме, че в отговора **не може да има цифрата 1** (освен в случая, когато n = 1 ).

Разлагането на числото ***K*** на прости множители може да съдържа само **прости числа по-малки от 10:** 2, 3, 5 и 7. С множителите **5 и 7** е лесно — те ще влязат в отговора като **отделни цифри.** Двойките и тройките обаче могат да се **групират,** така че броят на цифрите да намалее:

* Три двойки могат да се заменят с 8: ( 2^3 = 8 ).
* Две тройки могат да се заменят с 9: ( 3^2 = 9 ).

След това може да останат **не повече от две двойки** и **не повече от една тройка**.

* Ако остане една двойка и една тройка, те могат да се заменят с 6.
* Ако останат две двойки, те могат да се заменят с 4.
* Ако след това остане единична двойка или тройка, те ще се включат в отговора като **самостоятелни цифри.**

Накрая, получените цифри трябва да се **изведат в нарастващ ред** (от по-малките към по-големите).

Стигаме до следното решение — нека делим нашето число на всички възможни цифри от 9 до 2 в низходящ ред, т.е. ще се опитаме в произведението да „отделим“ колкото се може повече деветки, после колкото се може повече осмици и т.н. Ако в резултат на деленето от числото ***K*** остане 1, то трябва да изведем намерените цифри в обратен ред. Ако обаче остане число, по-голямо от 1, това означава, че първоначалното ***K*** е имало делител, по-голям от 10, и в такъв случай трябва да изведем „XXX“. Също така трябва отделно да обработим случая, когато ***K* =** 1 — тогава отговорът също е 1. В показаното по-долу решение това се случва, ако низът *ans* е празен, тоест ако за числото ***K*** не е намерен нито един едноцифрен делител.

int main() {

long long k;

cin >> k;

string ans = "";

for (int d = 9; d >= 2; d--) {

while (k % d == 0) {

k /= d;

ans = char('0' + d) + ans;

}

}

if (k == 1) {

if (ans.empty()) ans = "1";

cout << ans << endl;

} else {

cout << "XXX" << endl;

}

return 0;

}

Решение за най-малко 35 точки може да се получи, ако „изпробваме“ всички числа, започвайки от 1, докато не се намери такова число, чийто произведение на цифрите е равно на ***K***.

Пример за такова решение – agent\_25.cpp.

int prod(int n) {

int res = 1;

while (n > 0) {

res \*= n % 10;

n /= 10;

}

return res;

}

int main() {

int k;

cin >> k;

int ans = 1;

while (prod(ans) != k) {

ans++;

}

cout << ans << endl;

return 0;

}

Това решение не отчита факта, че може да липсва подходящо решение. Подобна идея е реализирана в agent\_54.cpp, но там се ограничава цикъла до 105.

*Автор: Кинка Кирилова-Лупанова*