

### Проблем 3. Производ цифара

Дат је цео број  $N$ . Исписати најмањи позитиван цео број  $X$  такав да је производ цифара броја  $X$  једнак броју  $N$ . Уколико такав број не постоји исписати  $-1$ .

**Улаз.** (Улазни подаци се учитавају на стандардног улаза) Први ред стандардног улаза садржи број  $0 \leq N \leq 10^{18}$ .

**Излаз.** (Излазне подаци се исписују на стандардни излаз) У првом реду стандардног излаза исписати број  $X$ .

#### Пример 1.

<code>proizvod.in</code>	<code>proizvod.out</code>
56	78

**Ограничења.** Временско ограничење: 0.1s; Меморијско ограничење: 16MB

---

**Решење и анализа.** При анализи проблема наилазимо на два дела: да ли представљање у траженом облику постоји и ако постоји потребно је наћи најмањи такав број. Размотримо прво услове за егзистенцију решења.

Сваки природан број може бити представљен на јединствен начин као производ простих бројева, до на поредак (пошто је множење комутативна операција односно  $a \cdot b = b \cdot a$ ). Овај диван резултат је позната као основна теорема алгебре. Како се од нас тражи да пронађемо број чији је производ цифара једнак датом броју  $N$  то значи да се он може представити као производ једноцифрених простих бројева. Заиста, уколико би прост број  $p > 10$  делио број  $N$  тада он не би могао бити изражен као производ бројева мањих од 10 односно производ цифара. Дакле, имамо да је  $N$  мора бити облика  $2^a \cdot 3^b \cdot 5^c \cdot 7^d$  да би тражени број  $X$  постојао.

Како најлакше испитати ово? Ограничење броја  $N$  је  $10^{18}$ , зато једноставним делењем редом са 2, 3, 5 односно 7, све док је то могуће долазимо до неке нове вредности  $N$ . На крају, број  $N$  мора завршити на јединицу. У супротном, тражено представљање не постоји и као резултат враћамо  $-1$ .

У тексту проблема се захтева да број  $X$  буде најмањи могућ. Како ово постићи? Наиме, како посматрамо цифре број  $X$  прво што желимо да минимизујемо јесте управо број цифара. Како знамо њихов производ, минимални број цифара добијамо уколико за сваку од њих бирамо управо највећу могућу вредност. Други речима, минимални број цифара форме  $X$  добијамо тако што природни број  $N$  делимо са највећом цифром, односно са 9, све док је то могуће. Затим прелазимо на цифру 8 и све тако до цифре 2 (цифру 1 не треба посматрати, јер она само повећава број цифара, а не утиче на производ). На овај начин се добија минимални број цифара траженог број  $X$ .

На овај начин смо одредили минимални број цифара траженог броја  $X$ . Поставља се питање како наћи цифре овог броја. Цифре управо представљају описани делиоци при трежењу број цифара. Другим речима, ако смо број  $N$  поделили 5 пута цифром 9, тада ће тражени број  $X$  имати управо пет цифара 9 у свом запису. Оно што преостаје јесте да се ове цифре и сортирају. Сортирањем желимо да обезбедимо да цифре веће тежине имају мање вредности,

чиме добијамо најмању вредност броја  $X$ .

Горе описано сортирање је најбоље одрадити пребројавањем (енг. *count sort*). Другим речима, дефинишемо низ *count* дужине 9 и иницијализујмо вредности на нула. Када делимо број  $N$  цифром  $k$ , повећавамо бројач *count*[ $k$ ] (он заправо представља неку врсту степена). На крају, сортирани низ цифара изгледа као: *count*[1] пута се појављује јединица, затим *count*[2] пута број два, све до *count*[9] пута број девет. Ово је најједноставнији алгоритам сортирања, међутим он се може применити само уколико је скуп вредности елемената низа мали и дискретан (као у ово случају где их има само 9).

Међутим, како ми редом и испитујемо цифре, нема потребе за сортирањем. Тачније, ми налазимо цифре почев од највеће, тако да је само потребно обрнути поредак. Ово можемо урадити на крају или при самој иницијализацији цифара.

Комплетан код у *Pascal*-у је дат у наставку.

```
program ProizvodCifara;
var
  c: Integer;
  n: QWord;
  solution: String;
begin
  ReadLn(n);
  if (n = 0) then begin
    Writeln(10);
    Exit;
  end;
  if (n = 1) then begin
    Writeln(1);
    Exit;
  end;
  solution := '';
  for c := 9 downto 2 do begin
    while (n > 1) and (n MOD c = 0) do begin
      n := n DIV c;
      solution := c + solution;
    end;
  end;
  if (n <> 1) then
    Writeln(-1);
  else
    Writeln(solution);
end.
```