

Podzielność iloczynu (rozwiązanie)

Autor zadania: Adam Gąsienica-Samek
Opracowanie: Filip Konieczny, Karolina Kulas
Opis rozwiązania: Bartosz Kostka



Zadanie

W zadaniu należy sprawdzić, czy iloczyn N podanych liczb całkowitych jest podzielny przez 6.

Rozwiązanie

Kluczową obserwacją jest fakt, że liczba jest podzielna przez 6 wtedy, gdy jest podzielna zarówno przez 2, jak i przez 3. Wynika to z faktu, że $6 = 2 \cdot 3$, a czynniki 2 i 3 są liczbami pierwszymi.

Aby iloczyn liczb $A_1 \cdot A_2 \cdot \dots \cdot A_n$ był podzielny przez 2, co najmniej jedna z liczb A_i musi być podzielna przez 2 (czyli parzysta). Podobnie, aby iloczyn był podzielny przez 3, co najmniej jedna z liczb A_i musi być podzielna przez 3.

Łącząc te dwa warunki, dochodzimy do wniosku, że iloczyn wszystkich podanych liczb jest podzielny przez 6 wtedy i tylko wtedy, gdy w ciągu znajduje się **co najmniej jedna liczba parzysta** oraz **co najmniej jedna liczba podzielna przez 3**. Należy zwrócić uwagę, że jedna liczba podzielna przez 6 (np. 12) spełnia oba te warunki jednocześnie.

Rozwiązanie sprowadza się do wczytania wszystkich liczb i sprawdzenia, czy powyższe warunki są spełnione. Możemy użyć dwóch zmiennych logicznych (lub typu `int` przyjmujących wartości 0 lub 1), które będą przechowywać informację, czy znaleźliśmy już liczbę podzielną przez 2 oraz liczbę podzielną przez 3.

Przechodzimy przez wszystkie N liczb. Dla każdej wczytanej liczby a sprawdzamy:

- czy $a \% 2 == 0$? Jeśli tak, odnotowujemy, że znaleźliśmy liczbę parzystą.
- czy $a \% 3 == 0$? Jeśli tak, odnotowujemy, że znaleźliśmy liczbę podzielną przez 3.

Po sprawdzeniu wszystkich liczb, jeśli oba warunki zostały spełnione, wypisujemy TAK. W przeciwnym wypadku wypisujemy NIE.

pod.cpp

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N;
8      cin >> N;
9      bool czy_znaleziona_parzysta = false,
10         czy_znaleziona_podzielna_przez_3 = false;
11      for (int i = 0; i < N; ++i)
12      {
13          int A;
14          cin >> A;
15          if (A % 2 == 0)
16              czy_znaleziona_parzysta = true;
17          if (A % 3 == 0)
18              czy_znaleziona_podzielna_przez_3 = true;
19      }
20      if (czy_znaleziona_parzysta and czy_znaleziona_podzielna_przez_3)
21          cout << "TAK\n";
22      else
23          cout << "NIE\n";
24  }
```



```
1 N = int(input())
2 As = map(int, input().split())
3
4 czy_znaleziona_parzysta = False
5 czy_znaleziona_podzielna_przez_3 = False
6
7 for A in As:
8     if A % 2 == 0:
9         czy_znaleziona_parzysta = True
10    if A % 3 == 0:
11        czy_znaleziona_podzielna_przez_3 = True
12
13 if czy_znaleziona_parzysta and czy_znaleziona_podzielna_przez_3:
14     print("TAK")
15 else:
16     print("NIE")
```

Oba rozwiązania działają w złożoności czasowej $\mathcal{O}(N)$, ponieważ każdą liczbę analizujemy tylko raz. Złożoność pamięciowa jest stała ($\mathcal{O}(1)$), ponieważ ilość używanej pamięci nie rośnie wraz z liczbą wczytywanych danych.¹

Błędne rozwiązania

Warto zwrócić uwagę, że próba obliczenia pełnego iloczynu i sprawdzenia jego podzielności przez 6 zawiedzie dla dużych danych. Iloczyn liczb może bardzo szybko przekroczyć zakres nawet 64-bitowych typów danych (np. `long long` w C++), co prowadzi do błędnych wyników. Takie rozwiązanie zadziała tylko dla pierwszego podzadania, gdzie liczby są małe.

¹Drobna uwaga jest taka, że w rozwiązaniu w języku Python wczytujemy wszystkie liczby i trzymamy je w pamięci, co de facto daje nam złożoność $\mathcal{O}(N)$. Gdybyśmy wczytywali liczby jedna po drugiej, wtedy moglibyśmy mieć stałą złożoność pamięciową, aczkolwiek znacznie obniżyłoby to czytelność kodu.