

## Zadanie «Numerowanie budynków» (num)

W pewnym mieście postanowiono ponumerować budynki, które się w nim znajdują. Budynkom zostaną nadane kolejne numery będące wartościami całkowitymi począwszy od 1. W tym celu zostały zamówione specjalne tabliczki, gdzie na każdej tabliczce znajduje się wyłącznie jedna cyfra. Przykładowo, aby nadać numery dziewięciu budynkom konieczne jest posiadanie dziewięciu tabliczek. Z kolei nadanie numerów dziesięciu budynkom wymaga jedenastu tabliczek (numer 10 to dwie tabliczki).

Dla problemu nadawania numerów można zadać następujące pytania: ilu budynkom zostaną nadane numery, jeżeli wiemy, ile zostało zamówionych tabliczek? Czy liczba zamówionych tabliczek jest poprawna? Liczba zamówionych tabliczek jest poprawna wyłącznie w przypadku, kiedy po nadaniu numerów budynkom nie pozostaną wolne tabliczki. Przykładowo, zamówienie dwunastu tabliczek jest błędne, ponieważ w taki sposób można nadać numery dziesięciu budynkom i jedna tabliczka pozostanie wolna.

### Specyfikacja wejścia

Wejście zawiera wiele zestawów danych testowych. Każdy zestaw danych składa się z jednej linii zawierającej jedną liczbę całkowitą  $n$  ( $1 \leq n \leq 2\,000\,000\,000$ ) będącą liczbą zamówionych tabliczek.

Dane wejściowe zakończone są wierszem zawierającym liczbę 0 (znacznik końca danych wejściowych, nie są wykonywane dla niego żadne obliczenia).

### Specyfikacja wyjścia

Program powinien wypisać dla każdego zestawu jedną liczbę, będącą liczbą budynków którym zostaną nadane numery lub wypisać słowo **Niepoprawne!**, jeżeli liczba zamówionych tabliczek jest błędna.

### Przykładowe wejście

```
101
192
32
12493
0
```

### Przykładowe wyjście

```
55
100
Niepoprawne!
3400
```

### Nazewnictwo plików

Dane wejściowe znajdują się w pliku o nazwie `num-01.in`, plik ten zawiera wiele zestawów danych wejściowych. Dane wyjściowe (rozwiązania dla wszystkich zestawów danych wejściowych) powinny zostać zapisane do jednego pliku wyjściowego o nazwie `num-01.out`.

### Ustalenia techniczne

1. Rozwiązaniem zadania są:

- program konsolowy napisany w języku C/C++ – Kod źródłowy programu powinien być zawarty wyłącznie w jednym pliku o nazwie `num.c` (dla języka C) lub `num.cpp` (dla języka C++). W pierwszej linii pliku źródłowego należy umieścić w komentarzu *indywidualny kod uczestnika* (IKU). Nie jest dopuszczalne umieszczanie w kodzie jakichkolwiek innych danych umożliwiających zidentyfikowanie uczestnika (także we właściwościach pliku).
- plik wyjściowy `num.out` wypracowany przez program dla danych testowych dostarczonych wraz z treścią zadania – Plik musi być nazwany zgodnie z niżej umieszczonym nazewnictwem. Plik musi być zgodny ze specyfikacją wyjścia.

Wszystkie powyższe pliki należy spakować do pliku `IKU-num.zip`, gdzie IKU jest *indywidualnym kodem uczestnika*.

2. Program powinien odczytywać dane wejściowe z pliku o nazwie podanej w treści zadania, a wynik należy zapisać też do pliku, którego nazwa jest podana w treści zadania.
3. Należy przyjąć, że dane wejściowe mają poprawny format (opisany w treści zadania). Plik wyjściowy powinien mieć format opisany w treści zadania.
4. W programach można korzystać wyłącznie ze standardowych bibliotek języka C/C++.
5. W programach nie można korzystać z rozwiązań i mechanizmów nieprzenośnych (np. zależnych od systemu operacyjnego).
6. Programy nie mogą:
  - tworzyć nowych procesów lub wątków,
  - uruchamiać innych programów,
  - używać funkcji sieciowych (np. `socket`, `send` itp.),
  - oczekiwać na interakcję użytkownika.
7. Zadanie należy przesłać przez stronę konkursu «Złoty Indeks» Platformy Zdalnej Edukacji korzystając z łącza do przesyłania rozwiązań zadania «num».
8. Zadanie jest oceniane w skali 0-15 punktów.