

Sumujący Jaś

Dostępna pamięć: 32 MB.

Może to zabrzmieć dziwnie, ale Jaś uwielbia pisemne rachunki. Jaś jest niezmordowany, a koledzy i koleżanki prześcigają się w wymyślaniu dla niego kolejnych przykładów. Nowym wyzwaniem będzie sumowanie liczb zapisanych w postaci ułamków dziesiętnych. W czasie, gdy Jaś będzie pracować nad sumowaniem, napisz program, który sprawdzi poprawność jego obliczeń.

Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia liczbę,
- obliczy ich sumę,
- wypisze wynik na standardowe wyjście.

Wejście

Pierwszy wiersz wejścia zawiera liczbę całkowitą N ($1 \leq N \leq 100$) oznaczającą ilość liczb dziesiętnych do zsumowania, zapisanych w kolejnych N wierszach, po jednej liczbie w wierszu.

Liczba składa się ze znaku ('+' lub '-'), po którym następuje od 1 do 100 cyfr, przecinek i od 1 do 100 cyfr po przecinku. Gdy liczba jest nieujemna można w jej zapisie pominąć znak '+'. Gdy liczba jest całkowita można pominąć przecinek i cyfry na prawo od niego.

Wyjście

W pierwszym wierszu wyjścia należy wypisać obliczoną sumę w zapisie dziesiętnym. Spośród wielu możliwych zapisów odpowiedzi należy wybrać najkrótszy.

Przykład

Dla danych wejściowych:

```
3
12,4
+13,9
-20
```

poprawnym wynikiem jest:

```
6,3
```

Rozwiązanie

Jak łatwo zauważyć, rozmiar liczb wczytywanych do programu (do 200 cyfr) znacznie przekracza pojemność dowolnej zmiennej liczbowej wbudowanej w dostępne języki programowania. Musimy więc zaimplementować własną arytmetykę obsługującą duże liczby.

Przypomnijmy sobie metody dodawania i odejmowania pisemnego. Zapisujemy dwie liczby w kolejnych wierszach, wyrównując je do prawej strony (cyfry jedności obu liczb znajdują się jedna nad drugą, tak jak cyfry dziesiątek itd.). Następnie przesuwamy się od prawej do lewej, sumując cyfry w kolejnych kolumnach. Ponieważ w każdej kolumnie może znajdować się tylko jedna cyfra, to wyniki dwucyfrowe należy obcinać do cyfry jedności (co odpowiada operacji wzięcia reszty z dzielenia przez 10), a do następnej kolumny dodać jedynekę.

Odejmowanie przebiega bardzo podobnie, istotna różnica występuje w momencie uzyskania w kolumnie ujemnego wyniku odejmowania dwóch cyfr. W takim wypadku należy wziąć “pożyczkę” — w aktualnej kolumnie do wyniku dodajemy 10, a z następnej odejmujemy jedynekę.

Istotą rozwiązania jest zaimplementowanie algorytmu działania pisemnego. Możemy wobec tego potraktować wczytywane liczby jako ciągi cyfr, które umieścimy w odpowiednich kolumnach (np. komórkach tablicy) i będziemy wykonywać na nich odpowiednie operacje.

Dla wygody implementacyjnej warto wyrównać obie liczby do jednakowej długości znakowej, dostawiając dodatkowe zera z lewej lub prawej strony, pamiętając oczywiście o konieczności wypisania wyniku bez nieznaczących zer.

Można jeszcze ułatwić odejmowanie większej liczby od mniejszej przez odwrócenie problemu. Zauważmy bowiem, że dla dwóch dowolnych liczb x, y wynik operacji $|x - y|$ jest równy $|y - x|$. Czyli jeśli mamy policzyć $x - y$, gdzie $x < y$, to zamiast tego możemy policzyć $y - x$ i zmienić znak otrzymanego wyniku. Aby uniknąć konieczności wielokrotnego zamieniania znaków w wypadku naprzemiennego dodawania i odejmowania posumujemy najpierw wszystkie liczby dodatnie, z każdej liczby ujemnej usuwamy znak minus i także dodajemy je do siebie, a następnie odejmijmy od pierwszej liczby drugą.