

Zadanie «Stary kalkulator» (sta)

Tomek przeglądając rodzinne pamiątki znalazł stary kalkulator. Kalkulator ten wykonuje jedynie cztery podstawowe operacje (dodawanie, odejmowanie, mnożenie i dzielenie) działając na liczbach całkowitych (wykonuje dzielenie całkowitoliczbowe) maksymalnie czterocyfrowych z zakresu $0, \dots, 9999$. Po włączeniu wyświetlacz kalkulatora jest „pusty” (nie wyświetla żadnej liczby). Poza dziesięcioma przyciskami odpowiadającymi cyfrom i czterema odpowiadającymi operacjom arytmetycznym (dodawanie ‘+’, odejmowanie ‘-’, mnożenie ‘*’ i dzielenie ‘/’), kalkulator posiada jeszcze przycisk ‘=’, który oblicza i wyświetla wynik (naciśnięcie przycisku operacji arytmetycznej nie powoduje obliczenia i wyświetlenia wyniku). Jeżeli wynik operacji nie mieści się w zakresie $0, \dots, 9999$, to w wyniku naciśnięcia ‘=’ nie zostanie wyświetlona żadna liczba.

Tomka zaciekało, jak wiele różnych liczb jest w stanie wyświetlić na kalkulatorze z wykorzystaniem jedynie niektórych przycisków kalkulatora.

Specyfikacja wejścia

Wejście zawiera wiele zestawów danych testowych. Każdy zestaw składa się z dwóch linii. W pierwszej linii znajduje się liczba K ($0 < K \leq 15$) oznaczająca liczbę wykorzystywanych przycisków. W drugiej linii znajduje się K różnych znaków wypisanych bez odstępu oznaczających wybrane przez Tomka przyciski. Możliwe znaki należą do zbioru: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, =\}$.

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 ilością wszystkich liczb, które Tomek może wyświetlić na kalkulatorze za pomocą wybranych przycisków.

Przykładowe wejście

```
5
3=7+*
1
1
2
1+
3
+=1
3
+2=
0
```

Przykładowe wyjście

```
9993
4
4
9999
4999
```

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 `sta.c` (dla języka C) lub `sta.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 `sta.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-sta.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 «sta».
8. Zadanie jest oceniane w skali 0-15 punktów.

Nazewnictwo plików

Dane wejściowe znajdują się w pliku o nazwie `sta.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 `sta.out`.