

**Zadanie:**

W pliku tekstowym z danymi znajdują się odpowiednio zapisane działania arytmetyczne do wykonania, gdzie każde działanie składa się z operacji, systemu liczbowego, w jakim podane są liczby oraz argumenty (o maksymalnej długości 40, max długość wykładnika potęgowania - 10). Należy stworzyć plik wynikowy zawierający dane działania i ich wyniki.

Dopuszczalne operacje to:

- dodawanie,
- mnożenie,
- potęgowanie,
- dzielenie całkowitoliczbowe,
- dzielenie modulo
- konwersja systemu liczbowego

Dopuszczalne są systemy liczbowe o podstawie z zakresu od 2 do 16.

**Plik wejściowy:**

Tekstowy plik wejściowy zawiera listę działań arytmetycznych, które program ma wykonać. Działania arytmetyczne zapisane są w formacie: znak działania, spacja, podstawa systemu argumentów, lista argumentów oddzielonych pustymi liniami. W przypadku konwersji w linii działania zapisane są: podstawa systemu argumentów, spacja, podstawa systemu docelowego. Poszczególne działania rozdzielone są trzema liniami przerwy.

```
<znak działania> <podstawa systemu argumentów>

<argument 1>

<argument 2>

<argument 3>


<podstawa systemu argumentów> <podstawa systemu docelowego>

<argument 1>
```

**Znaki działań:**

+ dodawanie

\* mnożenie

/ dzielenie całkowitoliczbowe

% dzielenie modulo

^ potęgowanie

Plik wyjściowy:

W pliku wyjściowym zapisywane są podane na wejściu działania wraz z wstawionym wynikiem.

```
+ 16
A3
10
B3
^ 8
63
7
15036166036133
% 2
01000011
10101
100
10 16
980
3D4
```

### Implementacja i uruchomienie:

Program napisany w języku C.

Uruchamiany w linii poleceń konsoli z pliku \*.exe.

Działający w systemie Windows.

Program wczytuje dane z pliku wejściowego i zapisuje wyniki w pliku wyjściowym. Nazwy plików podawane są jako argumenty wywołania. Jeśli nazwa pliku wyjściowego nie zostanie podana to program stworzy ją na podstawie nazwy pliku wejściowego.

Format nazwy pliku wejściowego: \*.txt

Format nazwy pliku wyjściowego: out\_\*.txt

Uruchomienie:

```
calc.exe <ścieżka do pliku wejściowego> [ścieżka do pliku wyjściowego]
```

Program powinien odpowiednio radzić sobie z błędami, np. z błędnym formatem pliku wejściowym, czy błędnymi danymi w tym pliku. Powinien np. wykonać tyle działań, na ile jest to możliwe.

### Dokumentacja:

Kod powinien być odpowiednio okomentowany. Dodatkowo należy przygotować dokumentację z opisem implementacji.