

Zadanie α

Teodozjusz

Jesteś informatykiem pracującym w firmie Bajtomania. Obecne Twoje zlecenie złożył szalony profesor Teodozjusz Malkontent. Zajmuje się on astronomią i wykonuje całą masę różnych obliczeń (właściwie nie wiadomo po co...). Naukowiec potrzebuje programu do wykonywania operacji na macierzach.

Operacje do zaimplementowania zdefiniowane są następująco:

- ADD - dodawanie dwóch macierzy podanych w kolejnych liniach.
- MULTIPLY - mnożenie dwóch macierzy podanych w kolejnych liniach.
- POWER p - podnosi do potęgi p macierz podaną w kolejnych liniach.

Dla szybszego działania programu podczas potęgowania macierzy należy zastosować **algorytm szybkiego potęgowania**.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą z ($1 \leq z \leq 2 \cdot 10^9$) – liczbę operacji do wykonania. Opis jednej operacji jest następujący:

W pierwszej linii zestawu znajduje się nazwa operacji, a w przypadku mnożenia dodatkowo liczba naturalna p ($1 \leq p \leq 10^6$).

W drugiej linii znajdują się dwie liczby naturalne n_1 oraz m_1 ($1 \leq n_1, m_1 \leq 100$) oznaczające rozmiar pierwszej macierzy. W kolejnych n_1 liniach znajduje się $n_1 * m_1$ liczb, którymi należy wypełnić pierwszą macierz.

Jeśli operacją nie jest POWER w kolejnej linii zestawu znajdują się dwie liczby naturalne n_2 oraz m_2 ($1 \leq n_2, m_2 \leq 100$) oznaczające rozmiar drugiej macierzy. W kolejnych n_2 liniach znajduje się $n_2 * m_2$ liczb, którymi należy wypełnić drugą macierz.

Wyjście

Dla każdej operacji wypisz wynik. Jeśli nie da się wykonać operacji wypisz słowo *ERROR*. Dla prostoty obliczeń wystarczy, jeśli podasz trzy ostatnie cyfry szukanych liczb (elementów macierzy).

Dostępna pamięć: 2MB

Wymagany język: C++

Zadanie należy zrealizować poprzez zaimplementowanie opisanej poniżej klasy `matrix`.

- Klasa posiada cztery pola prywatne: tablicę przechowującą macierz, pola przechowujące wymiary macierzy oraz pole `error` informujące o błędzie powstałym podczas wyznaczania macierzy. Dla uproszczenia obliczeń, w klasie przechowywane są tylko trzy ostatnie cyfry poszczególnych pól macierzy.
- Klasa posiada konstruktor z trzema parametrami: liczba wierszy, liczba kolumn, pole `error` będące informacją o błędzie. Wartościami domyślnymi parametrów powinny być: 0, 0, `false`.

```
matrix A(3,4);  matrix B(0,0,true);  matrix C;
```

- Klasa jest zaprzyjaźniona z pięcioma operatorami:
 1. `matrix operator+(const matrix& A, const matrix& B)` – zwraca wynik dodawania macierzy *A* i *B*.
 2. `matrix operator*(const matrix& A, const matrix& B)` – zwraca wynik mnożenia macierzy *A* i *B*.
 3. `matrix operator^(const matrix& A, int p)` – zwraca *A* do potęgi *p*.
 4. `ostream& operator<<(ostream &stream, const matrix &A)` – wypisuje macierz *A* do strumienia *stream*.
W przypadku, gdy macierz nie została poprawnie wyznaczona (pole `error == true`), operacja wypisuje słowo `ERROR` i znak końca linii.
 5. `istream& operator>>(istream &stream, matrix &A)` – wczytuje macierz *A* ze strumienia *stream*.

W przypadku niemożności wykonania operacji matematycznych, operatory `+`, `*`, `^` zwracają macierz z polem `error` ustawionym na `true`.

- Klasa może zawierać dodatkowe pola lub metody.

Definicję klasy wraz z definicjami wszystkich operatorów należy umieścić w pliku z rozszerzeniem `.h`. Tak przygotowany plik należy wysłać na Satori. Zostanie on skompilowany wraz z poniższym plikiem zawierającym funkcję `main`.

```
#include<iostream>
using namespace std;
#include "solution.h"

int main()
{
    ios_base::sync_with_stdio(false);
    string name;
    int n, m, p, z;

    cin >> z;
    while (z--)
    {
        cin >> name;
        if (name[0] == 'P') cin >> p;

        cin >> n >> m;
        matrix W;
        matrix A(n,m);
        cin >> A;

        switch (name[0])
        {
            case 'A' :
            {
                cin >> n >> m;
                matrix B(n,m);
                cin >> B;
                W = A+B;
                break;
            }
            case 'M' :
            {
                cin >> n >> m;
                matrix B(n,m);
                cin >> B;
                W = A*B;
                break;
            }
            case 'P' : W = A^p; break;
        }

        cout << W;
    }
    return 0;
}
```

Dostępna pamięć: 2MB
Wymagany język: C++

Przykład

Dla danych wejściowych:

```
6
ADD
1 1
3
1 1
5
ADD
3 2
1 2
3 4
1 1
3 2
3 3
3 3
2 2
MULTIPLY
2 3
1 1 1
1 1 2
3 3
3 3 1
3 3 1
2 2 1
MULTIPLY
2 2
1 1
1 1
3 2
3 3
3 3
3 3
POWER 4
2 2
3 3
3 3
POWER 8
2 3
1 2 1
3 1 2
```

Poprawną odpowiedzią jest:

```
8
4 5
6 7
3 3
8 8 3
10 10 4
ERROR
648 648
648 648
ERROR
```