

Zadanie I

Leniwy Jaś

Dziś w szkole, do której uczęszcza Jaś, na lekcji informatyki uczniowie poznawali różne systemy pozycyjne oraz sposób zapisywania liczb w pamięci komputera. Na zakończenie lekcji nauczycielka zadała całą masę przykładów do przeliczenia. Jaś po powrocie do domu spojrzał na te przykłady z wyraźną niechęcią. Przeliczenie tego wszystkiego będzie bardzo żmudne i nudne i zajmie mi całe popołudnie!!!

Po chwili namysłu Jaś postanowił napisać program komputerowy, który odrobi pracę domową za niego... Program będzie obsługiwał dwa rodzaje poleceń:

- *CONVERT* p q *liczba* - przekształca podaną liczbę naturalną z systemu o podstawie p na liczbę w systemie o podstawie q .
- *WRITE* r x - przekształca podaną liczbę całkowitą x (w systemie dziesiętnym) na jej reprezentację w rejestrze o rozmiarze r w kodzie uzupełnieniowym do 2. Jeśli liczba nie mieści się w rejestrze należy wypisać *ERROR*. Uwaga: liczba może być ujemna.
- *READ* *rejestr* - wyznacza wartość dziesiętną liczby x zapisanej w kodzie uzupełnieniowym do 2 w rejestrze *rejestr*.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą z ($1 \leq z \leq 2 \cdot 10^9$) – liczbę zestawów danych, których opisy występują kolejno po sobie. Opis jednego zestawu jest następujący:

Każdy zestaw składa się z jednego z trzech możliwych poleceń opisanych wyżej, w których $2 \leq p, q \leq 16$, $4 \leq r \leq 64$, natomiast długość liczby jest ograniczona przez 60. Dla podstawy systemu większej niż 10, kolejnymi cyframi są A, B, C, D, E, F .

Możesz założyć, że wartość dziesiętna wszystkich liczb pojawiających się w zadaniu nie przekracza 10^{18} .

Wyjście

Dla każdego polecenia w jednej linii wypisz uzyskaną liczbę (w odpowiednim systemie) lub reprezentację rejestru w postaci dwójkowej.

Wersja I1 - obsługuje tylko polecenie *CONVERT*, wersja za 0.5 pkt

Wersja I2 - obsługuje tylko polecenia *WRITE*, *READ*, wersja za 0.5 pkt

Dostępna pamięć: 2MB

Wymagany język: C

Przykład

Dla danych wejściowych:

```
14
CONVERT 10 2 7
CONVERT 2 10 1010
CONVERT 2 8 101011
CONVERT 3 9 12012
CONVERT 5 8 12340
CONVERT 2 16 101001
CONVERT 16 2 BC
WRITE 8 12
WRITE 8 -12
WRITE 8 -128
WRITE 8 -130
READ 00001100
READ 01111111
READ 11111111
```

Poprawną odpowiedzią jest:

```
111
10
53
165
1712
29
10111100
00001100
11110100
10000000
ERROR
12
127
-1
```