

## Zadanie Y Wielomiany

Ostatnimi czasy szalony naukowiec Teodozjusz wykonuje ogromne ilości operacji na wielomianach dużych stopni. Przydałby mu się kolejny program. Zawitał więc znów do Twojej firmy składając kolejne zlecenie.

Twoim zadaniem jest napisać klasę `doubleList`, będącą **wskaźnikową implementacją listy podwójnie wiązanej** reprezentującej wielomian. Lista powinna być posortowana rosnąco wg stopni jednomianów wchodzących w skład wielomianu. Dodatkowo, zalecana jest implementacja listy z "głową" i "ogonem" lub listy cyklicznej z wartownikiem.

Na wielomianach będą wykonywane następujące operacje:

- `PRINT_ASC id` - wypisanie wielomianu o identyfikatorze `id` w porządku rosnących stopni. Jeśli wielomian jest zerowy, funkcja wypisuje `EMPTY`.
- `PRINT_DESC id` - wypisanie wielomianu o identyfikatorze `id` w porządku malejących stopni. Jeśli wielomian jest zerowy, funkcja wypisuje `EMPTY`.
- `DERIVATIVE id` - obliczenie pochodnej wielomianu o identyfikatorze `id`. Wynikiem operacji jest wielomian o identyfikatorze `id`.
- `ADD_MONO id a b` - dodanie do wielomianu o identyfikatorze `id` jednomianu postaci  $b \cdot x^a$ . Jeśli po dodaniu jednomianu współczynnik przy  $x^a$  równy jest 0 element listy reprezentujący ten stopień należy z listy usunąć. Funkcja wypisuje `ADD OK`.
- `ADD id1 id2` - dodanie do siebie dwóch wielomianów o identyfikatorach `id1` oraz `id2`. Wynikiem dodawania jest wielomian o identyfikatorze `id1`. Wielomian o identyfikatorze `id2` pozostaje bez zmian. Jeśli po dodaniu któryś współczynnik równy jest 0, element listy reprezentujący ten współczynnik należy z listy usunąć. Funkcja wypisuje słowo `ADD OK`. Można założyć, że `id1` jest różny od `id2`.
- `MULTI_CONST id a` - mnożenie wielomianu o identyfikatorze `id` przez liczbę całkowitą `a` (która może być równa 0). Funkcja wypisuje `MULTI OK`.
- `MULTI id1 id2` - mnożenie przez siebie dwóch wielomianów o identyfikatorach `id1` oraz `id2`. Wynikiem mnożenia jest wielomian o identyfikatorze `id1`. Wielomian o identyfikatorze `id2` pozostaje bez zmian. Jeśli po pomnożeniu któryś współczynnik równy jest 0, element listy reprezentujący ten współczynnik należy z listy usunąć. Funkcja wypisuje `MULTI OK`. Można założyć, że `id1` jest różny od `id2`.
- `MIN id` - wypisanie współczynnika przy jednomianie o najmniejszym stopniu w wielomianie o identyfikatorze `id`; Jeśli wielomian jest zerowy, funkcja wypisuje `ERROR`.
- `MAX id` - wypisanie współczynnika przy jednomianie o największym stopniu w wielomianie o identyfikatorze `id`; Jeśli wielomian jest zerowy, funkcja wypisuje `ERROR`.

- `CLEAN id` - zerowanie wielomianu o identyfikatorze `id`; Funkcja wypisuje `CLEAN OK`, jeżeli operacja działa na niezerowym wielomianie; w przeciwnym razie wypisuje `EMPTY`.

## 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:

Pierwsza linia zawiera liczbę  $n$  oznaczającą liczbę operacji do wykonania na wielomianach ( $0 \leq n \leq 10^6$ ). Kolejne  $n$  linii zawiera: kod operacji oraz stosowne dla operacji argumenty oddzielone spacją. Argumentami są : identyfikator wielomianu (duża litera alfabetu angielskiego) oraz w zależności od operacji stopień (liczby naturalna z zakresu od 0 do 15000), współczynnik lub stała (są to liczby całkowite z zakresu od  $-1000$  do  $1000$ ). Uwaga: Liczby powstające podczas operacji mogą być duże (mieszczą się w `long long int`).

**Wersja Y1** - nie obsługuje poleceń `MULTI_CONST` oraz `MULTI`, wersja za 0.6 pkt.

**Wersja Y2\*** - obsługuje wszystkie polecenia, wersja za dodatkowe 0.4 pkt.

## Wyjście

Dla każdego zestawu danych wykonaj odpowiedni ciąg operacji. Wielomiany wypisz zgodnie z zamieszczonymi niżej przykładami. Uwaga: nie wypisuj współczynnika równego 1 oraz -1.

**Dostępna pamięć:** w zależności od testu 2-8MB

**Wymagany język:** C++

## Przykład

Dla danych wejściowych:

```
1
22
ADD_MONO A 1 1
ADD_MONO B 0 3
ADD_MONO B 1 -4
ADD_MONO B 6 1
MIN A
MAX B
ADD A B
PRINT_ASC A
DERIVATIVE B
PRINT_DESC B
ADD_MONO C 0 3
ADD_MONO C 1 -3
ADD_MONO C 6 1
MULTI_CONST C -1
ADD A C
PRINT_ASC A
MULTI B C
PRINT_ASC B
CLEAN B
ADD_MONO B 0 4
MULTI B C
PRINT_ASC B
```

Poprawną odpowiedzią jest:

```
ADD OK
ADD OK
ADD OK
ADD OK
1
1
ADD OK
 $3-3x^1+x^6$ 
 $6x^5-4$ 
ADD OK
ADD OK
ADD OK
MULTI OK
ADD OK
EMPTY
MULTI OK
 $12-12x^1-18x^5+22x^6-6x^{11}$ 
CLEAN OK
ADD OK
MULTI OK
 $-12+12x^1-4x^6$ 
```