Zadanie U: Dynamiczna tablica

Twoim zadaniem jest efektywna implementacja interfejsu dynamicznej tablicy. Interfejs zawiera operację dodawania i usuwania elementu na końcu, sprawdzania i przypisywania elementu pod wskazany indeks oraz inicjalizację tablicy o zadanej długości. Zakładamy, że na samym początku tablica nie posiada żadnych elementów.
Uwaga! Nie korzystaj z kontenera std::vector, ale wykorzystaj podobny mechanizm alokacji pamięci do tego, który używa ten kontener. Jest on opisany między innymi tutaj: `http://en.wikipedia.org/wiki/Dynamic_array`.

Format wejścia

Pierwsza linia wejścia zawiera liczbę całkowitą n, $0 \leq n \leq 10^6$ - ilość poleceń. W kolejnych n liniach znajduje się jedno polecenie z poniższych:

- $RESIZE\ n\ x$ - inicjalizacja tablicy o rozmiarze n, której każdy element ma ustaloną wartość x. Poprzednie wartości elementów są nadpisywane.
- $GET\ i$ - zwraca wartość elementu w tablicy pod indeksem i.
- $SET\ i\ x$ - ustawia wartość elementu w tablicy pod indeksem i na x.
- $PUSH\_BACK\ x$ - dodaje element x na koniec tablicy.
- POP_BACK - usuwa ostatni element tablicy.

Wartości powyższych zmiennych są ograniczone: $0 \leq n, i, x \leq 10^6$. Możesz założyć, że operacja POP_BACK nigdy nie zostanie wywołana na pustej tablicy a SET nigdy nie otrzyma indeksu, który nie istnieje w tablicy.

Format wyjścia

Na wyjściu dla każdej komendy GET powinien zostać wypisany wynik jej działania (wartość elementu o zadanym indeksie). Jeśli operacja GET zostanie wykonana na indeksie większym lub równym rozmiarowi tablicy, to program powinien wypisać napis $OUT\ OF\ BOUND$.

Przykład

Dla danych wejściowych:

```
9
PUSH_BACK 2
GET 0
GET 1
RESIZE 5 1
GET 4
SET 4 8
GET 4
POP_BACK
GET 4
```

Poprawną odpowiedzią jest

```
2
OUT OF BOUND
1
8
OUT OF BOUND
```