Vector

동적 배열인 벡터를 환형 배열로 구현하라. 이때, 다음과 같은 명령어들을 수행할 수 있어야 한다.

개발자가 벡터를 개발할 때 내부에 생성하는 벡터의 크기(용량)를 N이라고 하고, 사용자가 인식하는 크기를 n이라고 하자. **배열을 확장할 때는 항상 더블링 기법을 활용한다. 배열을 축소할 때는 N이 n/2보다 작거나 같을 경우(단, N이 2보다 작아지지 않는다) 축소한다.**

벡터의 첫 용량은 2이다.

1. at(i): i번째 값을 출력한다. 만약 해당하는 인덱스 값이 없는 경우(비어 있는 경우 포함) -1을 출력한다.
2. set(i, x): i번째 값을 x로 바꾼다. 이때 해당하는 인덱스 값이 없는 경우(비어 있는 경우 포함) -1을 출력한다.
3. **insert(i, x): i번째 값에 x를 넣는다. i는 [0, n]의 범위에서 주어진다. 이때, 벡터가 비어있는 경우 -1을 출력한다.**
4. **remove(i): i번째 값을 삭제하고, 삭제한 값을 출력한다. 이때 해당하는 인덱스 값이 없는 경우(비어 있는 경우 포함) -1을 출력한다.**
5. **push\_back(x): 배열의 맨 뒤에 x를 추가한다.**
6. getSize(): 사용자가 인식하는 배열의 크기 n을 출력한다.
7. empty(): 동적배열이 비어있는지 확인한다. 비어 있을 경우 1을, 아니면 0을 출력한다.
8. print(): 배열의 모든 원소를 공백을 사이에 두고 순차적으로 출력한다. 비어 있을 경우 -1을 출력한다.

입력

가장 첫 줄에는 명령어 입력의 수 N이 주어진다. N의 범위는 [0, 10,000]이다.

각 줄마다 명령어가 주어지며, 매개변수가 필요한 경우 공백으로 구분하여 주어진다.

이때, 벡터의 원소로 주어지는 숫자 x는 [0, 1,000] 사이의 정수이다.

|  |  |
| --- | --- |
| 30  empty  getSize  remove 2  print  push\_back 3  set 2 3  insert 0 1  insert 1 2  print  set 0 3  print  getSize  push\_back 4  push\_back 5  push\_back 6  push\_back 7  insert 4 9  print  getSize  empty  remove 3  remove 7  remove 0  remove 4  remove 5  remove 4  print  getSize  push\_back 6  print | 1  0  -1  -1  -1  1 2 3  3 2 3  3  3 2 3 4 9 5 6 7  8  0  4  -1  3  6  -1  7  2 3 9 5  4  2 3 9 5 6 |