B - Append Editorial by en_translator

This problem can be solved by appropriately using an array (list).

Below, we introduce sample code in Python, C++, and C. For most of the other languages, implementation would be similar to one of them.

In Python, one can insert x to the tail of a list A with `A.append(x)`, and obtain the x-th last element of a list A with `A[-x]`.

```python
Q=int(input())
A=[]
for _ in range(Q):
  t,x=map(int,input().split())
  if t==1:
    A.append(x)
  else:
    print(A[-x])
```

In C++, one can insert x to the tail of a vector A with `A.push_back(x)`, and obtain the x-th last element of a vector A with `A[A.size()-x]`.

```cpp
#include<bits/stdc++.h>
using namespace std;
int main(){
  int Q;
  cin >> Q;
  vector<int>A;
  for(int i=0;i<Q;i++){
    int t,x;
    cin >> t >> x;
    if(t==1){
      A.push_back(x);
    }else{
      cout << A[A.size()-x] << endl;
    }
  }
}
```

C does not have something like vector in C++. The simplest implementation would be initially reserving a fixed-length array of length exceeding the maximum possible number of elements, and managing the actual size by yourself.

Copy

```
#include<stdio.h>
int main(){
  int Q;
  scanf("%d",&Q);
  int A[101];
  int size=0;
  for(int i=0;i<Q;i++){
    int t,x;
    scanf("%d%d",&t,&x);
    if(t==1){
      A[size]=x;
      size++;
    }else{
      printf("%d\n",A[size-x]);
    }
  }
}
```