Cool vector   X90362_en

*To solve this problem you have to complete the code that you will find at the end of the statement. You have to replace each ??? with an expression of code. **Do not change anything else.** Download from the website of the problem the file code.cc with the code to be completed (click on the corresponding button ".CPP"), edit it and submit it to the judge. We also provide you with a file main.cc to help you when testing your solution, which you **must not** submit to the judge.*

In this problem, we say that a vector with n integer numbers $v[0..n-1]$ is *cool* if $n \geq 2$, $v[0] > v[n-1]$, and there exists an index j between 0 and $n-2$ such that:

- $v[0] \leq \ldots \leq v[j-1] \leq v[j]$,

- $v[j+1] \leq v[j+2] \leq \ldots \leq v[n-1]$.

For instance, the vector $[12, 12, 15, 20, 1, 3, 3, 5, 9]$ is cool (with $j = 3$).

Implement an *efficient* function

    int *search* (**int** x, **const** *vector* <**int**>& v);

that returns the position of the *last occurrence* of x in a cool vector v. If x does not belong to v, return a -1.

Precondition

The vector v is cool.

```
#include <iostream>
#include <vector>
using namespace std;

int position (const vector <int>& v, int e, int d) {
   if (e+1 == d) return e;
   int m = (e+d)/2;
   if (???) return position (v, m, d);
   else      return position (v, e, m);
}

int search (int x, const vector <int>& v, int e, int d) {
   if (e > d) return −1;
   if (e == d) return (v[e] == x ? e : −1);
   int m = ???   // Pay attention when d == e + 1
   if (???) return search (x, v, e, ???);
   else      return search (x, v, ???, d);
}

int search (int x, const vector <int>& v) {
   int n = v.size ();
```

```
      int j = position (v, 0, n−1);
      int p = search (x, v, 0, j );
      if (p ≠ −1) return p;
      return search (x, v, j+1, n−1);
   }
```

Problem information