```c
// file: bsearch-re.c
//
// Visualization (search for 2 as an example):
//  https://pythontutor.com/visualize.html#code=%23include%20%3Cstdio.h%3E%0A%0A%23define%20LEN%2010%0A%0Aint%20BinarySearch%28int%20key,%20const%20int%20dict%5B%5D,%20int%20low,%20int%20high%29%3B%0A%0Aint%20main%28%29%20%7B%0A%20%20const%20int%20dictionary%5BLEN%5D%20%3D%20%7B%200,%201,%201,%202,%203,%205,%208,%2013,%2021,%2034%20%7D%3B%0A%0A%20%20int%20key%20%3D%202%3B%0A%0A%20%20printf%28%22The%20index%20of%20%25d%20is%20%25d.%5Cn%22,%20key,%0A%20%20%20%20%20%20%20%20%20%20BinarySearch%28key,%20dictionary,%200,%20LEN%20-%201%29%29%3B%0A%0A%20%20return%200%3B%0A%7D%0A%0Aint%20BinarySearch%28int%20key,%20const%20int%20dict%5B%5D,%20int%20low,%20int%20high%29%20%7B%0A%20%20if%20%28low%20%3E%20high%29%20%7B%0A%20%20%20%20return%20-1%3B%0A%20%20%7D%0A%0A%20%20int%20mid%20%3D%20%28low%20%2B%20high%29%20/%202%3B%0A%0A%20%20if%20%28dict%5Bmid%5D%20%3D%3D%20key%29%20%7B%0A%20%20%20%20return%20mid%3B%0A%20%20%7D%0A%0A%20%20if%20%28dict%5Bmid%5D%20%3E%20key%29%20%7B%0A%20%20%20%20return%20BinarySearch%28key,%20dict,%20low,%20mid%20-%201%29%3B%0A%20%20%7D%0A%0A%20%20return%20BinarySearch%28key,%20dict,%20mid%20%2B%201,%20high%29%3B%0A%7D&cumulative=true&heapPrimitives=nevernest&mode=edit&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false
// Created by hfwei on 2023/11/9.

#include <stdio.h>

#define LEN 10

int BinarySearch(int key, const int dict[], int low, int high);

int main() {
  const int dictionary[LEN] = { 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 };

  int key;
  scanf("%d", &key);

  printf("The index of %d is %d.\n", key,
         BinarySearch(key, dictionary, 0, LEN - 1));

  return 0;
}

int BinarySearch(int key, const int dict[], int low, int high) {
//  if (low == high) {
//    if (dict[low] == key) {
//      return low;
//    }
//    return -1;
//  }

  if (low > high) {
    return -1;
  }
```

```
36
37   int mid = (low + high) / 2;
38
39   if (dict[mid] == key) {
40     return mid;
41   }
42
43   if (dict[mid] > key) {
44     return BinarySearch(key, dict, low, mid - 1);
45   }
46
47   return BinarySearch(key, dict, mid + 1, high);
48 }
```