```c
/*
*
* Implement linear search
* Peter Strawn
* CS50 AP
*
*/

#include <cs50.h>
#include <stdio.h>

#define LENGTH 10

int main(void)
{
    // declare array of 10 numbers
    int haystack[LENGTH] = { 9, 7, 3, 4, 2, 8, 1, 6, 0, 5 };

    // get needle to find in haystack
    printf("Tell me the integer you are looking for: ");
    int needle = GetInt();

    // iterate through list to find item
    for (int i = 0; i < LENGTH; i++)
    {
        // check if needle is found at index
        // use return to end program if necessary
        if (needle == haystack[i])
        {
            printf("Needle found at index %d!\n", i);
            return 0;
        }
        // if needle not found, increment i
    }

    // inform user that needle isn't in haystack
    printf("Needle not in haystack!\n");
    return 1;
}
```

```c
/*
*
* Implement binary search
* Peter Strawn
* CS50 AP
*
*/

#include <cs50.h>
#include <stdio.h>

#define LENGTH 10

int main(void)
{
    // declare array of 10 numbers
    int haystack[LENGTH] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

    // get needle to find in haystack
    printf("Tell me the integer you are looking for: ");
    int needle = GetInt();

    // divide and conquer to find it
    // declare minimum starting index as 0
    int min = 0;

    // declare maximum starting index as index of last item in list
    int max = LENGTH - 1;

    while (min <= max)
    {
        // locate midpoint index between min and max
        int mid = (min + max) / 2;

        // check if needle is at midpoint
        if (needle == haystack[mid])
        {
            printf("Needle found at %d!\n", mid);
            return 0;
        }
        // if needle is greater than midpoint, update minimum index
        else if (needle > haystack[mid])
        {
            min = mid + 1;
        }
        // if needle is less than midpoint, update maximum index
        else if (needle < haystack[mid])
        {
```

```
            max = mid - 1;
        }
    }

    // inform user that needle isn't in haystack
    printf("Needle not in haystack!\n");
    return 1;
}
```