Lucky Numbers

Lucky numbers are subset of integers. Rather than going into much theory, let us see the process of arriving at lucky numbers,

Take the set of integers
1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,……

First, delete every second number, we get following reduced set.
1,3,5,7,9,11,13,15,17,19,……

Now, delete every third number, we get
1, 3, 7, 9, 13, 15, 19,……..

Continue this process indefinitely……
Any number that does NOT get deleted due to above process is called “lucky”.

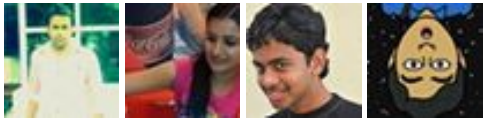
Therefore, set of lucky numbers is 1, 3, 7, 13,…….

Now, given an integer ‘n’, write a function to say whether this number is lucky or not.

```
    bool isLucky(int n)
```

Algorithm:
Before every iteration, if we calculate position of the given no, then in a given iteration, we can determine if the no will be deleted. Suppose calculated position for the given no. is P before some iteration, and each lth no. is going to be removed in this iteration, if P < l then input no is lucky, if P is such that P%l == 0 (l is a divisor of P), then input no is not lucky.

Recursive Way:



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```c
#include <stdio.h>
#define bool int

/* Returns 1 if n is a lucky no. ohterwise returns 0*/
bool isLucky(int n)
{
  static int counter = 2;

  /*variable next_position is just for readability of
     the program we can remove it and use n only */
  int next_position = n;
  if(counter > n)
    return 1;
  if(n%counter == 0)
    return 0;

 /*calculate next position of input no*/
  next_position -= next_position/counter;

  counter++;
  return isLucky(next_position);
}

/*Driver function to test above function*/
int main()
{
  int x = 5;
  if( isLucky(x) )
    printf("%d is a lucky no.", x);
  else
    printf("%d is not a lucky no.", x);
  getchar();
}
```

Example:

Let's us take an example of 19

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,15,17,18,19,20,21,……

1,3,5,7,9,11,13,15,17,19,…..

1,3,7,9,13,15,19,…….

1,3,7,13,15,19,……

1,3,7,13,19,……

In next step every 6th no .in sequence will be deleted. 19 will not be deleted after this step

Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding "extern" keyword in C

Median of two sorted arrays

Tree traversal without recursion and without stack!

Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

because position of 19 is 5th after this step. Therefore, 19 is lucky. Let's see how above C code finds out:

Current function call	Position after this call	Counter for next call	Next Call
isLucky(19)	10	3	isLucky(10)
isLucky(10)	7	4	isLucky(7)
isLucky(7)	6	5	isLucky(6)
isLucky(6)	5	6	isLucky(5)

When isLucky(6) is called, it returns 1 (because counter > n).

Iterative Way:
Please see this comment for another simple and elegant implementation of the above algorithm.

Please write comments if you find any bug in the given programs or other ways to solve the same problem.

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion
- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

Writing code in comment? Please use **ideone.com** and share the link here.

15 Comments GeeksforGeeks

Sort by Newest

Rohan · 9 months ago

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

bool isLucky(int n){

int i = 1;
int inc = 2;
while(i<=n){
if(n-i == 0)
return true;
i+= inc;
```


Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 17 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 57 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

inc+=2;
}
return false;;
}

int main() {
cout<<islucky(1000001)<<endl; return="" 0;="" }="">

Rohan • 9 months ago

This can also be tried...

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

bool isLucky(int n){

        int i = 1;
        int inc = 2;
        while(i<=n){
                if(n-i == 0)
                        return true;
                i+= inc;
                inc+=2;
        }
        return false;;
}



int main() {
```

```
        return 0;
}
```

Rahul Jujarey • a year ago

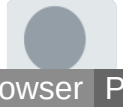
Wikipedia Article
http://en.wikipedia.org/wiki/L...

defines that we the counter should be next surviving number, and not increme

4

anon • 2 years ago

```
/* Paste your code here (You may delete these lines if not writing co
bool isLucky(int n)
{
        int pos = n;
        int del = 2;
        while( pos >= del ){
                if( pos%del == 0 )
                        return false;
                pos = pos - pos/del;
                del++;
        }
        return true;
}
```



Kanhaiya • 2 years ago

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

void lucky_num(int n)
{
int gap = 0;
int l_num = 0;
int i = 0;

while(i < n)
{
int c_gap = 0;
while(c_gap++ <= gap) i++;
cout<<i<<" ";
l_num++;
if(i == n)
{
```

see more

Kanhaiya Kanhaiya • 2 years ago

Assumption: we know that 1 is lucky. And want to test a number other

aravind • 2 years ago

Thanks a lot for good code...

TEST • 3 years ago

25 is not lucky no. as written above.

GeeksforGeeks ➦ TEST • 3 years ago

@TEST: Thanks for pointing this out. We have removed 25 from the lis

ravi • 3 years ago

this is iterative version .

```
void findlucky(int n)
{
   int i ;
   for(i=2;n &gt;=i ; i++ )
   {
      if(n%i == 0)
      {
         print &quot;unlucky&quot; ;
         return ;
      }
      n = n - n/i ;
  }
  print &quot;lucky&quot;
  return ;
}
```

manush • 4 years ago

Please give the source code and explanation for the Fibonacci series in an arr

@Saravanan Mani: Thanks very much, nice implementation. We have include

Saravanan Mani • 4 years ago

```
#include <stdio.h>
#define bool int

bool isLucky(int n)
{
  int next_pos = n, counter;

  for(counter=2; counter <= next_pos; ++counter)
  {
    if(next_pos%counter == 0)
      return 0;

    /* Next position is (current position –
      no of positions reduced in this iteration) */
    next_pos -= next_pos/counter;
  }
```

see more

geeksforgeeks • 5 years ago

@Minjie Zha: Thanks a lot for pointing out the issue, we have fixed it. Count sh position is calculated. Keep it up!!

I think there is some wrong with the recursive part, I test the function with 5, ar
should be:
return isLucky(n-n/(counter-1));

Am I right?

Subscribe Add Disqus to your site