**6 kyu**

**Life without primes**

18092% of 7929 of214[KenKamau](https://www.codewars.com/users/KenKamau)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/life-without-primes/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/life-without-primes/solutions/cpp)

[Forks (3)](https://www.codewars.com/kata/life-without-primes/forks/cpp)

[Discourse (38)](https://www.codewars.com/kata/life-without-primes/discuss/cpp)

* Add to Collection
* |
* Share this kata:

Consider an array that has no prime numbers, and none of its elements has any prime digit. It would start with: [1,4,6,8,9,10,14,16,18,..]. The element at index 1 is 4.

12 and 15 are not in the list because 2 and 5 are primes.

You will be given an integer n and your task will be return the number at that index in the array. For example, solve(1) = 4, as explained above.

More examples in the test cases.

Good luck!

If you like Prime Katas, you will enjoy this Kata: [Simple Prime Streaming](https://www.codewars.com/kata/5a908da30025e995880000e3)

<https://www.codewars.com/kata/life-without-primes/cpp>

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

bool es\_primo(int n) {

    if(n < 2) return false;

    if(n == 2) return true;

    if(n%2 ==0) return false;

    for(int i =3; i\*i<=n; i+=2){

        if(n%i==0) return false;

    }

    return true;

}

int solve(int n) {

  //your code here

  std::vector<int> v;

  v.push\_back(1);

  v.push\_back(4);

  int  cont =2;

  for(int i = 5; cont < n + 1; i++) {

      if(!es\_primo(i)) {

          int copia = i;

          bool flag = true;

          while(copia > 0) {

              int d = copia % 10;

              if(d == 2 || d == 3 || d == 5 || d == 7) {

                  flag = false;

                  break;

              }

              copia /= 10;

          }

          if(flag) {

                v.push\_back(i);

                cont++;

          }

    }

  }

*/\**

*for(int i =0; i<v.size(); i++) {*

*cout << v[i] << " ";*

*}\*/*

  //return v[n];

  return v[n];

}

//iskoch

*/\**

*iskoch*

*int next(int n) {*

*if (n % 10 == 0 || n % 10 == 8) return n + 1;*

*if (n % 10 == 1) return n + 3;*

*if (n % 10 == 4 || n % 10 == 6) return n + 2;*

*return next(n / 10) \* 10;*

*}*

*bool isPrime(int n) {*

*if (n == 1) return false;*

*for (int i = 2; i\*i <= n; i++) {*

*if (n % i == 0) return false;*

*}*

*return true;*

*}*

*int solve(int n) {*

*int num = 1;*

*for (int i = 0; i < n; i++) {*

*do {*

*num = next(num);*

*} while (isPrime(num));*

*}*

*return num;*

*}*

*\*/*

int main() {

    return 0;

}