How to check if a given number is Fibonacci number?

Given a number 'n', how to check if n is a Fibonacci number.

We strongly recommend that you click here and practice it, before moving on to the solution.

A simple way is to generate Fibonacci numbers until the generated number is greater than or equal to 'n'. Following is an interesting property about Fibonacci numbers that can also be used to check if a given number is Fibonacci or not.

A number is Fibonacci if and only if one or both of $(5*n^2 + 4)$ or $(5*n^2 - 4)$ is a perfect square (Source: Wiki). Following is a simple program based on this concept.

```
// C++ program to check if x is a perfect square
#include <iostream>
#include <math.h>
using namespace std;
// A utility function that returns true if x is perfect square
bool isPerfectSquare(int x)
    int s = sqrt(x);
    return (s*s == x);
}
// Returns true if n is a Fibinacci Number, else false
bool isFibonacci(int n)
{
    // n is Fibinacci if one of 5*n*n + 4 or 5*n*n - 4 or both
    // is a perferct square
    return isPerfectSquare(5*n*n + 4) ||
           isPerfectSquare(5*n*n - 4);
// A utility function to test above functions
int main()
  for (int i = 1; i <= 10; i++)
     isFibonacci(i)? cout << i << " is a Fibonacci Number \n":
                     cout << i << " is a not Fibonacci Number \n";</pre>
  return 0;
}
```

Run on IDE

Output:

```
1 is a Fibonacci Number
2 is a Fibonacci Number
3 is a Fibonacci Number
4 is a not Fibonacci Number
5 is a Fibonacci Number
6 is a not Fibonacci Number
7 is a not Fibonacci Number
8 is a Fibonacci Number
9 is a not Fibonacci Number
10 is a not Fibonacci Number
```