Exercise Fibonacci Numbers

- Fibonacci numbers are defined recursively as:
  - fibo(n) = 1, when n=0 or n=1
  - fibo(n) = fibo(n-1) + fibo(n-2) when n>1
- **Complete the skeleton code** that computes the n-th fibonacci number **recursively**.
- Given an input integer n greater than or equal to 0, display the n-th fibonacci number. Do **not** use any loops!
- Test case 1 :

  Input:

  ```
  10
  ```

  Output:

  ```
  89
  ```

Exercise Fibonacci Numbers Skeleton Code

```
#include <stdio.h>
// write the prototype below


// complete the main function to read input, call functions, and display output
int main() {


    return 0;
}
// complete the function below
int fibo(int n) {
    // base case


    // recursive step


}
```

you can copy paste this skeleton code into Codecast, complete and test it there, and then copy paste to ***complete the skeleton code*** in LMO Exercise autograder

WARNING: Hints to the exercise on the next slide

Please try to solve the exercise by yourself first...

Exercise Fibonacci Numbers Hints

- Add the prototype, don't forget the semicolon
- In main,
  - read the input number
  - printf the fibonacci number number,
    while calling the function on the input (in printf)
- In fibo,
  - Base case:
    if n is 0 or 1, immediately return 1
    (must be both, otherwise you can run into infinite recursion)
  - Recursive step:
    call fibo on both n-1 and n-2, add them, return the result
    (can all be done in one return statement)