

Forum:[Hack Forums](#)

Thread: [display number in sequence C#](#)

unknown321 - December 5, 2019, 10:49 am

I have this code but it needs tweaking. I need to display a number in the Fibonacci sequence after entering a number in a textbox.

```
Code namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnCalculate_Click(object sender, EventArgs e)
        {
            int N = int.Parse(txtNumber.Text);

            double d2 = 0;
            double d1 = 1;
            double fib = d1 + d2;

            for (int i = 4; i < N; i++)
            {
                d2 = d1;
                d1 = fib;
                fib = d1 + d2;
            }

            txtFibo.Text = fib.ToString();
        }
    }
}
```

akali - December 5, 2019, 1:01 pm

```
Code - (int)fibonacci:(int)number{
    if(number <=1)
        return number;
    return [self fibonacci:number - 1 ] + [self fibonacci:number - 2];
}
-(void)main{
    NSLog(@"%i",[self fibonacci:9]);
}
```

realsunjester - December 5, 2019, 8:45 pm

(Dec 5, 2019 11:49 AM)unknown321 Wrote: [\[CITE\]](#) I have this code but it needs tweaking. I need to display a number in the Fibonacci sequence after entering a number in a textbox.

```

Code namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnCalculate_Click(object sender, EventArgs e)
        {
            int N = int.Parse(txtNumber.Text);

            double d2 = 0;
            double d1 = 1;
            double fib = d1 + d2;

            for (int i = 4; i < N; i++)
            {
                d2 = d1;
                d1 = fib;
                fib = d1 + d2;
            }

            txtFibo.Text = fib.ToString();
        }
    }
}

```

Code using System;

```

public class Program
{
    public static void Main()
    {
        Console.WriteLine(calcFib(13));
    }

    public static string calcFib(int i)
    {
        double ret = i * (1 + Math.Sqrt(5)) / 2.0;
        return Math.Round(ret).ToString();
    }
}

```

Gorzoid - December 7, 2019, 12:41 pm

(Dec 5, 2019 09:45 PM)realsunjester Wrote: [\[CITE\]](#) Code public static string calcFib(int i)

```

{
    double ret = i * (1 + Math.Sqrt(5)) / 2.0;
    return Math.Round(ret).ToString();
}

```

Did you even test this, fib(13) is 233 not 21 I am guessing you meant to use [this closed expression](#)

OP this is the C/C++ subforum, it's not for C# questions. Please ask your question in [the C# subforum](#). Taking a quick look your issue is likely due to a spurious 4 as your starting value in the for loop, maybe take a look at C# implementations of fibonacci on rosetta code.

0xDEAD10CC - December 7, 2019, 2:56 pm

```
Code private static Dictionary<int, ulong> _fib_cache = new Dictionary<int, ulong>();
```

```
private static ulong Fib(int n)
{
    if (n < 0 || n > 93)
        throw new ArgumentOutOfRangeException("Input must be greater than or equal to 0 and less than 94.");

    if (_fib_cache.ContainsKey(n))
        return _fib_cache[n];

    if (n < 2) return (ulong)n;
    var value = Fib(n - 1) + Fib(n - 2);

    _fib_cache[n] = value;
    return value;
}
```

More of a naive example, but you can use recursion and memoization for this.

realsunjester - December 7, 2019, 4:33 pm

(Dec 7, 2019 01:41 PM)Gorzoid Wrote: [\[CITE\]](#) Did you even test this, fib(13) is 233 not 21 I am guessing you meant to use [this closed expression](#)

OP this is the C/C++ subforum, it's not for C# questions. Please ask your question in [the C# subforum](#). Taking a quick look your issue is likely due to a spurious 4 as your starting value in the for loop, maybe take a look at C# implementations of fibonacci on rosetta code. seems to work just fine.

<https://dotnetfiddle.net/aFBiT1>

21 is the next number in the sequence after 13.

0xDEAD10CC - December 7, 2019, 7:02 pm

(Dec 7, 2019 05:33 PM)realsunjester Wrote: [\[CITE\]](#) seems to work just fine.

<https://dotnetfiddle.net/aFBiT1>

21 is the next number in the sequence after 13.

You don't understand what he's saying. Fib(13) != 21. You just proved that your code thinks Fib(13) is 21. The typical function Fib(N) usually implies that N is the Nth term of the fibonacci sequence. So your code doesn't work...

<https://www.wolframalpha.com/input/?i=Fib%2813%29>

realsunjester - December 7, 2019, 8:43 pm

(Dec 7, 2019 08:02 PM)0xDEAD10CC Wrote: [\[CITE\]](#) You don't understand what he's saying. Fib(13)

!= 21. You just proved that your code thinks Fib(13) is 21. The typical function Fib(N) usually implies that N is the Nth term of the fibonacci sequence. So your code doesn't work...

<https://www.wolframalpha.com/input/?i=Fib%2813%29>

clearly i misunderstood what he wanted. i thought he wanted the number that was next in the sequence, which is what it looked like he wanted.

but just so we are clear, my code does work, its just not what he wanted.

0xDEAD10CC - December 7, 2019, 9:51 pm

(Dec 7, 2019 09:43 PM)realsunjester Wrote: [\[CITE\]](#) clearly i misunderstood what he wanted. i thought he wanted the number that was next in the sequence, which is what it looked like he wanted.

but just so we are clear, my code does work, its just not what he wanted.

Well, not really:

```
Code // calcFib(0) = 0
```

```
// calcFib(1) = 2
```

Where's 1?

realsunjester - December 7, 2019, 10:07 pm

(Dec 7, 2019 10:51 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Well, not really:

```
Code // calcFib(0) = 0
```

```
// calcFib(1) = 2
```

Where's 1?

most fib sequences start at 2 since the first two are obviously 1 and 0. i would think anyone who would be using a fib would already know that. however if you feel you need some math theory in your life you can always turn to stack exchange, im sure youll find your argument there

Shisho - December 7, 2019, 10:39 pm

(Dec 7, 2019 11:07 PM)realsunjester Wrote: [\[CITE\]](#) most fib sequences start at 2 since the first two are obviously 1 and 0. i would think anyone who would be using a fib would already know that. however if you feel you need some math theory in your life you can always turn to stack exchange, im sure youll find your argument there

Already down-voted his question and marked as a duplicate!

0xDEAD10CC - December 8, 2019, 12:35 pm

(Dec 7, 2019 11:07 PM)realsunjester Wrote: [\[CITE\]](#) most fib sequences start at 2 since the first two are obviously 1 and 0. i would think anyone who would be using a fib would already know that. however if you feel you need some math theory in your life you can always turn to stack exchange, im sure youll find your argument there

No they don't start at 2, and they never did... :/ The sequence is well-defined and starting it from 2 would defeat the whole point proposed by the relationship between all of the numbers since starting at 2 wouldn't make sense. This isn't about what people know, this is about the fact that your function doesn't do what OP is trying to do, and it also doesn't work in the way you think it does.

<https://math.stackexchange.com/a/2300072>

Code Let $f_0=0, f_1=1, f_2=1, \dots$ be the Fibonacci numbers

<https://www.wolframalpha.com/input/?i=fib%280%29>

<https://www.wolframalpha.com/input/?i=fib%281%29>

<https://www.wolframalpha.com/input/?i=fib%282%29>

Having an arithmetic function that is intended to calculate a numerical value return a string also

makes absolutely no sense. C# is a strongly typed language for a reason.

If you were attempting to write Binet's formula:

<https://mobtowers.com/2015/08/21/calculating-the-nth-fibonacci-number-c/> (though he was even closer than you even with the OBOE in his code for higher terms.)

I forgot to mention, but I should've added overflow checks to my example as well. (edit: done, although changing to the decimal type would allow for larger calculations.)

```
Code using System;
using System.Collections.Generic;
```

```
public class Program
{
    public static void Main()
    {
        for (int i = 0; i <= 139; i++)
            Console.WriteLine("Fib({0}) = {1}", i, Fib(i));
    }

    private static Dictionary<int, decimal> _fib_cache = new Dictionary<int, decimal>();

    private static decimal Fib(int n)
    {
        if (n < 0 || n > 139)
            throw new ArgumentOutOfRangeException("Input must be greater than or equal to 0 and less than 139.");

        if (_fib_cache.ContainsKey(n))
            return _fib_cache[n];

        if (n < 2) return (ulong)n;
        var value = Fib(n - 1) + Fib(n - 2);

        _fib_cache[n] = value;
        return value;
    }
}

Code Fib(0) = 0
Fib(1) = 1
Fib(2) = 1
Fib(3) = 2
Fib(4) = 3
Fib(5) = 5
Fib(6) = 8
Fib(7) = 13
Fib(8) = 21
Fib(9) = 34
Fib(10) = 55
Fib(11) = 89
Fib(12) = 144
Fib(13) = 233
Fib(14) = 377
```

Fib(15) = 610
Fib(16) = 987
Fib(17) = 1597
Fib(18) = 2584
Fib(19) = 4181
Fib(20) = 6765
Fib(21) = 10946
Fib(22) = 17711
Fib(23) = 28657
Fib(24) = 46368
Fib(25) = 75025
Fib(26) = 121393
Fib(27) = 196418
Fib(28) = 317811
Fib(29) = 514229
Fib(30) = 832040
Fib(31) = 1346269
Fib(32) = 2178309
Fib(33) = 3524578
Fib(34) = 5702887
Fib(35) = 9227465
Fib(36) = 14930352
Fib(37) = 24157817
Fib(38) = 39088169
Fib(39) = 63245986
Fib(40) = 102334155
Fib(41) = 165580141
Fib(42) = 267914296
Fib(43) = 433494437
Fib(44) = 701408733
Fib(45) = 1134903170
Fib(46) = 1836311903
Fib(47) = 2971215073
Fib(48) = 4807526976
Fib(49) = 7778742049
Fib(50) = 12586269025
Fib(51) = 20365011074
Fib(52) = 32951280099
Fib(53) = 53316291173
Fib(54) = 86267571272
Fib(55) = 139583862445
Fib(56) = 225851433717
Fib(57) = 365435296162
Fib(58) = 591286729879
Fib(59) = 956722026041
Fib(60) = 1548008755920
Fib(61) = 2504730781961
Fib(62) = 4052739537881
Fib(63) = 6557470319842
Fib(64) = 10610209857723
Fib(65) = 17167680177565
Fib(66) = 27777890035288
Fib(67) = 44945570212853

Fib(68) = 72723460248141
Fib(69) = 117669030460994
Fib(70) = 190392490709135
Fib(71) = 308061521170129
Fib(72) = 498454011879264
Fib(73) = 806515533049393
Fib(74) = 1304969544928657
Fib(75) = 2111485077978050
Fib(76) = 3416454622906707
Fib(77) = 5527939700884757
Fib(78) = 8944394323791464
Fib(79) = 14472334024676221
Fib(80) = 23416728348467685
Fib(81) = 37889062373143906
Fib(82) = 61305790721611591
Fib(83) = 99194853094755497
Fib(84) = 160500643816367088
Fib(85) = 259695496911122585
Fib(86) = 420196140727489673
Fib(87) = 679891637638612258
Fib(88) = 1100087778366101931
Fib(89) = 1779979416004714189
Fib(90) = 2880067194370816120
Fib(91) = 4660046610375530309
Fib(92) = 7540113804746346429
Fib(93) = 12200160415121876738
Fib(94) = 19740274219868223167
Fib(95) = 31940434634990099905
Fib(96) = 51680708854858323072
Fib(97) = 83621143489848422977
Fib(98) = 135301852344706746049
Fib(99) = 218922995834555169026
Fib(100) = 354224848179261915075
Fib(101) = 573147844013817084101
Fib(102) = 927372692193078999176
Fib(103) = 1500520536206896083277
Fib(104) = 2427893228399975082453
Fib(105) = 3928413764606871165730
Fib(106) = 6356306993006846248183
Fib(107) = 10284720757613717413913
Fib(108) = 16641027750620563662096
Fib(109) = 26925748508234281076009
Fib(110) = 43566776258854844738105
Fib(111) = 70492524767089125814114
Fib(112) = 114059301025943970552219
Fib(113) = 184551825793033096366333
Fib(114) = 298611126818977066918552
Fib(115) = 483162952612010163284885
Fib(116) = 781774079430987230203437
Fib(117) = 1264937032042997393488322
Fib(118) = 2046711111473984623691759
Fib(119) = 3311648143516982017180081
Fib(120) = 5358359254990966640871840

Fib(121) = 8670007398507948658051921
Fib(122) = 14028366653498915298923761
Fib(123) = 22698374052006863956975682
Fib(124) = 36726740705505779255899443
Fib(125) = 59425114757512643212875125
Fib(126) = 96151855463018422468774568
Fib(127) = 155576970220531065681649693
Fib(128) = 251728825683549488150424261
Fib(129) = 407305795904080553832073954
Fib(130) = 659034621587630041982498215
Fib(131) = 1066340417491710595814572169
Fib(132) = 1725375039079340637797070384
Fib(133) = 2791715456571051233611642553
Fib(134) = 4517090495650391871408712937
Fib(135) = 7308805952221443105020355490
Fib(136) = 11825896447871834976429068427
Fib(137) = 19134702400093278081449423917
Fib(138) = 30960598847965113057878492344
Fib(139) = 50095301248058391139327916261
BigInteger if you wanted to go even further..