

IDLE_tmp_kcmme_

"""

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

"""

```
a=1
b=2
sum_even=2
n=2
while b<4000000:
    n=n+1
    b,a=a+b,b
    if b%2==0:
        sum_even=sum_even+b

print 'a=',a
print 'b=',b

print 'which one=',n
print 'Total of the even num=',sum_even
```

"""

```
the answers are
a= 3524578
b= 5702887
which one= 33
Total of the even num= 4613732
"""
```

"""

Here is a listing for maple

```
fib := proc(n)
local a,b,count,d; a := 1; b := 2; count := 2;
while a + b < n do
d := a + b; a := b; b := d;
if type(b,even) then count := count + b end if;
end do;
return(count);
```

IDLE_tmp_kcmme_

```
end proc;  
fib(1000000);
```

type in maple let's you check if a number is odd, even, prime, etc

Thinking about it, the mod function could be used to replace the if then:
count := count + b * (b+1 mod 2);

"""

"""

This may be a small improvement. The Fibonacci series is:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610...

Now, replacing an odd number with O and an even with E, we get:

O, O, E, O, O, E, O, O, E, O, O, E, O, O, E...

And so each third number is even. We don't need to calculate the odd numbers.
Starting from an two odd terms x, y, the series is:

x, y, x + y, x + 2y, 2x + 3y, 3x + 5y

And in Python, my solution is:

Python

Hide Code

```
def calcE():  
    x = y = 1  
    sum = 0  
    while (sum < 1000000):  
        sum += (x + y)  
        x, y = x + 2 * y, 2 * x + 3 * y  
    return sum
```

With this, you don't need to use an % to calculate if a number is even. Using a fast computer, this took less than 0.01 secs to calculate.

"""

```
"""
```

```
here simple code
```

```
(fibo: odd,odd,even,odd,odd,even,odd.....)
```

```
even = []
```

```
two, three = 1, 2
```

```
while two <= 4000000:
```

```
    even.append(three)
```

```
    one, two, three = two + three, two + 2*three, 2*two + 3*three
```

```
print(sum(even))
```

```
"""
```

```
"""
```

```
def fib(n,j,lim):
```

```
    if(j>lim):
```

```
        if((n%2 == 0)):
```

```
            return n
```

```
        else:
```

```
            print n,j
```

```
            return 0
```

```
    else:
```

```
        if((n%2 == 0)):
```

```
            return n + fib(j,n+j,lim)
```

```
        else:
```

```
            return fib(j,n+j,lim)
```

```
"""
```

```
"""
```

```
from itertools import takewhile
```

```
def fib():
```

```
    fk,fk1 = 0,1
```

```
    while True:
```

```
        yield fk
```

```
        fk, fk1 = fk1, fk + fk1
```

```
sum(filter(lambda x: x%2==0, list(takewhile(lambda x: x < 4*10**6, fib()))))
```

```
"""
```

```
"""
```

```
a, b = 0, 1
while b < 4000000:
    a, b = b, a+b
numbers = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
           1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393,
           196418, 317811, 514229, 832040, 1346269, 2178309, 3524578]
numbers2 = []
for num in numbers:
    if num % 2 == 0:
        numbers2.append(num)

x = numbers2
y = sum(numbers2)

print (y)
"""
```