

Fibonacci Strings (due 09 Oct 2020)

A Fibonacci string is a sequence of bit strings defined as follows:

$$\begin{aligned} f(n) &= '0' \text{ if } n = 0 \\ f(n) &= '1' \text{ if } n = 1 \\ f(n) &= f(n - 1) + f(n - 2) \text{ if } n > 1 \end{aligned}$$

The '+' operator denotes string concatenation.

Here are the first 6 terms in the series

n	f(n)
0	0
1	1
2	10
3	101
4	10110
5	10110101

Input: You will read your data from standard input. The format of the input will be as follows: There will be two lines of input. The first line will have an integer n where $0 \leq n \leq 30$. The second line of input will have a bit string p where $1 \leq \text{length of } p \leq 20$ characters. The length of p may exceed the length of the bit string returned by the fibonacci function.

Output: You will output a single integer that will be the number of times the bit string p occurs in $f(n)$. The occurrences of p may overlap.

Here are a couple of test cases:

n	p	num occurrences
6	101	4
20	1011101	0
30	1011	317811

The file (Fibonacci.py) that you will be submitting will have the following structure. You may **NOT** change the names of the functions but you may add as many helper functions as needed. You will follow the [standard coding conventions](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/assgn10.html) in Python.

```
# File: Fibonacci.py

# Description:

# Student's Name:

# Student's UT EID:

# Partner's Name:

# Partner's UT EID:

# Course Name: CS 313E

# Unique Number:

# Date Created:
```

```

# Date Last Modified:

import sys

# Input: n a positive integer
# Output: a bit string
def f ( n ):

# Input: s and p are bit strings
# Output: an integer that is the number of times p occurs in s
def count_overlap (s, p):

def main():
    # read n and p from standard input
    n = sys.stdin.readline()
    n = int (n.strip())
    p = sys.stdin.readline()
    p = p.strip()

    # compute the bit string f(n)

    # determine the number of occurrences of p in f(n)

    # print the number of occurrences of p in f(n)

if __name__ == "__main__":
    main()

```

You can always add more functions than those listed. We strongly recommend that you use an efficient implementation of the fibonacci function either using iteration or using a memo. We will adhere to the time limit set.

Let us say you have an input file *fib.in*:

```

8
1010

```

Then you can run your program on the command line:

```
python3 Fibonacci.py < fib.in
```

For this assignment you may work with a partner. Both of you must read the paper on [Pair Programming](#) and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will submit the code. Make sure that in the header you have your name and UT EID and your partner's name and UT EID. If you are working alone then you will just have your name and your UT EID.

Use the [Canvas](#) system to submit your **Fibonacci.py** file. We should receive your work by 11 PM on Friday, 09 Oct 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

- Your Python program should have the proper header.
- Your code must run before submission.
- You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.