

1. In mathematics, the Fibonacci numbers, commonly denoted F_n , form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1:

$$F_0 = 0, \quad F_1 = 1,$$

and

$$F_n = F_{n-1} + F_{n-2},$$

for $n > 1$

The beginning of the sequence is this: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

The function `fastFib(num)` returns the fibonacci number F_n , of the given num as an argument.

Examples:

`fib_fast(5)` → 5

`fib_fast(10)` → 55

`fib_fast(20)` → 6765

`fib_fast(50)` → 12586269025

Ans:

```
In [16]: 1 def fib_fast(num):
          2     x=0
          3     output = 0
          4     y = 1
          5     if num == 0:
          6         return 0
          7     elif num == 1:
          8         return 1
          9     else :
          10         for i in range(2, num+1):
          11             output = x+y
          12             x = y
          13             y = output
          14         print(f'fib_fast({num}) → {output}')
          15 fib_fast(5)
          16 fib_fast(10)
          17 fib_fast(20)
          18 fib_fast(50)

fib_fast(5) → 5
fib_fast(10) → 55
fib_fast(20) → 6765
fib_fast(50) → 12586269025
```

2. Create a function that takes a strings characters as ASCII and returns each characters hexadecimal value as a string.

Examples:

`convert_to_hex("hello world")` → "68 65 6c 6c 6f 20 77 6f 72 6c 64"

`convert_to_hex("Big Boi")` → "42 69 67 20 42 6f 69"

`convert_to_hex("Marty Poppinson")` → "4d 61 72 74 79 20 50 6f 70 70 69 6e 73 6f 6e"

Ans:

```
In [2]: 1 def convert_to_hex(string):
2         output = ''
3         for i in range(len(string)):
4             output += hex(ord(string[i]))[2:] + ' '
5         print(f'convert_to_hex({string})→{output}')
6 convert_to_hex("hello world")
7 convert_to_hex("Big Boi")
8 convert_to_hex("Marty Poppinson")

convert_to_hex(hello world→68 65 6c 6c 6f 20 77 6f 72 6c 64
convert_to_hex(Big Boi→42 69 67 20 42 6f 69
convert_to_hex(Marty Poppinson→4d 61 72 74 79 20 50 6f 70 70 69 6e 73 6f 6e
```

3. Someone has attempted to censor my strings by replacing every vowel with a *, I*k* th*s. Luckily, I've been able to find the vowels that were removed.

Given a censored string and a string of the censored vowels, return the original uncensored string.

Examples:

uncensor("Wh*r* d*d my v*w*ls g*?", "eeioeo") → "Where did my vowels go?"

uncensor("abcd", "") → "abcd"

uncensor("*PP*RC*S*", "UEAE") → "UPPERCASE"

Ans:

```
In [5]: 1 def uncensor(string,vowels):
2         n = 0
3         output = ''
4         for i in string:
5             if i == '*':
6                 output += vowels[n]
7                 n +=1
8             else:
9                 output += i
10        print(f'uncensor{string,vowels} → {output}')
11 uncensor("Wh*r* d*d my v*w*ls g*?", "eeioeo")
12 uncensor("abcd", "")
13 uncensor("*PP*RC*S*", "UEAE")

uncensor('Wh*r* d*d my v*w*ls g*?', 'eeioeo') → Where did my vowels go?
uncensor('abcd', '') → abcd
uncensor('*PP*RC*S*', 'UEAE') → UPPERCASE
```

4. Write a function that takes an IP address and returns the domain name using PTR DNS records.

Examples:

get_domain("8.8.8.8") → "dns.google"

get_domain("8.8.4.4") → "dns.google"

Ans:

```
In [9]: 1 def get_domain(ip_address):
2         import socket
3         result=socket.gethostbyaddr(ip_address)
4         print(f'get_domain({ip_address}) → {list(result)[0]}')
5 get_domain("8.8.8.8")
6 get_domain("8.8.4.4")
```

```
get_domain(8.8.8.8) → dns.google
get_domain(8.8.4.4) → dns.google
```

5. Create a function that takes an integer n and returns the factorial of factorials. See below examples for a better understanding:

Examples:

fact_of_fact(4) → 288

$4! * 3! * 2! * 1! = 288$

fact_of_fact(5) → 34560

fact_of_fact(6) → 24883200

Ans:

```
In [16]: 1 def fact_of_fact(num):
2         output = 1
3         import math
4         for i in range(num+1):
5             output *= math.factorial(i)
6         print(output)
7 fact_of_fact(4)
8 fact_of_fact(5)
9 fact_of_fact(6)
```

```
288
34560
24883200
```