Sum of possitive numbers

Calculate the sum of all possitive numbers in a list.
Example: for the list [2, -1, 3], the expected sum is 2 + 3 = 5

Entrée [1]:
```
a = [2, -1, 3]
# YOUR CODE HERE
length = len(a)
listSum = 0

for i in range(length):
    if a[i]>0:
        listSum += a[i]

print(f"Sum of list is {listSum}.")
# Hint:
# - Loop through elements of the list
# - Use conditional statement to check if an element is possitive
# - If yes, add to the final sum
```

```
Sum of list is 5.
```

Now, use the code that you've developed to create a function that takes in a list and returns the sum of positive numbers.

Entrée [2]:
```
def sum_positives(a_list):
    # YOUR CODE HERE
    listSum = 0
    for i in range(len(a_list)):
        if a_list[i]>0:
            listSum += a_list[i]
    return listSum
```

Call the function with the list `a` above.

Entrée [3]:
```
print(sum_positives(a))
```

```
5
```

Fibonacci

The Fibonacci (https://en.wikipedia.org/wiki/Fibonacci_number) series starts with 0 and 1. The next number is the sum of the last two numbers.

$x_0 = 0, x_1 = 1, x_{n+1} = x_n + x_{n-1}$

Write a function `get_Fibonacci_number` to compute x_n of the Fibonacci series. E.g:

- `get_Fibonacci_number(0)` returns 0
- `get_Fibonacci_number(1)` returns 1
- `get_Fibonacci_number(3)` returns 2

Entrée [4]:
```
def get_Fibonacci_number(n):
    # YOUR CODE HERE

    Fib0 = 0
    Fib1 = 1
    ListFib=[Fib0, Fib1]
    i=2
    Fibnext=0

    if n==0:
        return Fib0
    elif n==1:
        return Fib1
    else:
        while n>i-1:
            Fibnext=ListFib[i-1]+ListFib[i-2]
            ListFib.append(Fibnext)
            i+=1
        return ListFib[i-1]
```

Entrée [5]:
```
print(get_Fibonacci_number(0))
print(get_Fibonacci_number(1))
print(get_Fibonacci_number(3))
print(get_Fibonacci_number(10))
```

```
0
1
2
55
```

Write a function to get the largest Fibonacci number that is equal or smaller than a given number.
For example:

- Given 2, the functions should return 2
- Given 10, the functions should return 8

Entrée [6]:
```
# YOUR CODE HERE
def get_closest_Fibonacci_number(n):
    i=0
    while get_Fibonacci_number(i)<=n:
        i+=1
    return get_Fibonacci_number(i-1)
```

Entrée [7]:
```
print(get_closest_Fibonacci_number(2))
print(get_closest_Fibonacci_number(10))
```

```
2
8
```

Dictionary

A Python ditionary comprises of student numbers as keys and student names as values. Write a function to capitalize all the student names in the dictionary.

Entrée [8]:
```
# YOUR CODE HERE
def capitalize(dict):
```

```
    for i in dict:
        dict[i]=dict[i].capitalize()
    return dict
```

Entrée [9]:
```
dict1 = {"1": "sayerie", "2": "ravid"}
print(capitalize(dict1))
```

```
{'1': 'Sayerie', '2': 'Ravid'}
```

Character counts

Write a function that count the frequencies of each alphabet character in a given string. The function should return a dictionary, in which each key is a character and each value is the corresponding frequency. All characters are treated as their lowercases, meaning 'E' is the same as 'e'.
For example: Calling the function for 'Hello' will return {'h': 1, 'e': 1, 'l': 2, 'o': 1}.

Entrée [10]:
```
# YOUR CODE HERE
def freq(str):
    dictionary={}
    #str = sorted(str.lower())
    str=str.lower()
    for j in str:
        if j.isalpha():
            if j in dictionary:
                dictionary[j] = dictionary[j]+1
            else:
                dictionary.update({j:1}) #créer nouvelle ligne dans dictionnaire
    return dictionary
```

Entrée [11]:
```
print(freq('Hello'))
```

```
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

Extrema (Optional)

Given a list of numbers representing a series, count how many time the values change their trends, i.e. from increasing to descreasing and vi versa.

Examples of these changes are:

- [0, 2, 1]
- [0, -2, -2, 3]

Entrée [12]:
```
def inverse(list):
    change = 0
    plusplus =0
    minmin=0
    up=0
    down=0
    for i in (list):
        if list[i]>list[i-1]:
            up=1
            if plusplus!= up:
                change+=1
                plusplus=1
                minmin=0

        if list[i]<list[i-1]:
            down=1
            if minmin!= down:
                change+=1
                plusplus=0
                minmin=1
    return change
```

Entrée [13]:
```
a=[0, 2, 1]
print(inverse(a))

b=[0, -2, -2, 3]
print(inverse(b))
```

```
2
2
```

Approximate π (Optional)

One method to approximate the value of π is through simulation. Given the function `random` generates a number in the range $[0, 1]$ randomly, write a function to approxmiate π.

Hints:

- π is the area of a cirle with radius of 1.
- For any random point in the unit square (positions top-right of the origin), the change of this point belonging to the quarter unit circle is $\pi/4$

Entrée [14]:
```
from random import random
import math

def appro_pi(n):
    hit=0
    for i in range(n):
        x = random()
        y = random()
        if math.sqrt(x**2+y**2)<1:
            hit+=1
    pi=hit/(n)*4
    return pi
```

Entrée [15]:
```
print(appro_pi(10000000))
```

```
3.1418908
```

Entrée [ ]: