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

```
a = [2, -1, 3]
# YOUR CODE HERE
# Hint:
# - Loop through elements of the list
# - Use conditional statement to check if an element is possitive
# - If yes, add to the final sum
res = 0
for element in a:
    if element > 0:
        res = res + element
print(res)
```

```
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.

```
def sum_positives(a_list):
    # YOUR CODE HERE
    res = 0
    for element in a:
        if element > 0:
            res = res + element
    return res
```

```
12
```

Call the function with the list `a` above.

Fibonacci

The Fibonacci 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

```
def get_Fibonacci_number(n):
    # YOUR CODE HERE
    x0 = 0
    x1 = 1
    if n==0:
        return 0
    elif n>0:
        list = []
        list.append(1)
        list.append(1)
        for i in range(n):
            if i > 1:
                list.append(list[i-2]+list[i-1])
        return list[n-1]
    else:
        return "input error"
print(get_Fibonacci_number(3))
```

```
2
```

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

```
# YOUR CODE HERE
def get_largest_Fibonacci_number(n):
    list = []
    list.append(0)
    list.append(1)
    for i in range(n):
        while list[len(list)-1]<n:
            list.append(list[i-1]+list[i-2])
    return list[len(list)-2]
```

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.

```
# YOUR CODE HERE
student_info = {
    '123':'Williams',
    '124':'George',
    '125':'Steven'
}
def Capitalize_student_name(dict):
    for key in dict:
        dict[key] = dict.get(key).upper()
    return dict
print(Capitalize_student_name(student_info))
```

```
{'123': 'WILLIAMS', '124': 'GEORGE', '125': 'STEVEN'}
```

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}.

```
# YOUR CODE HERE
def count_characters(str):
    import collections
    dict = collections.defaultdict(int)
    str = str.lower()
    for c in str:
        dict[c] += 1
    return dict
text = 'hello world'
count_characters(text)
```

```
defaultdict(int,
            {'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 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]

```
def count_trend_changes(list):
    count = int()
    for i in range(2,len(list)):
        a = list[i-1] - list [i-2]
        b = list[i] - list[i-1]
        if a*b < 0:
            count += 1
        elif a*b == 0:
            if a+b !=0:
                count += 1
    return count
list = [0,-2,6,6,6,-3,2]
print(count_trend_changes(list))
```

```
4
```

Approximate $\pi$ (Optional)

```
from random import random
count = 0
i = 0
while i < 10000000:
    x = random()
    y = random()
    if pow(x-0.5,2)+pow(y-0.5,2)<0.25:
        count += 1
    i += 1
print(count*4/10000000)
```

```
3.1415768
```

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

Hints:

- $\pi$ 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 $\pi/4$