Programming_Assingment19

Question1

Create a function that takes a string and returns a string in which each character is repeated

once.

Examples

double_char('String') ➞ 'SSttrriinngg'

double_char('Hello World!') ➞ 'HHeelllloo WWoorrlldd!!'

double_char('1234!_ ') ➞ '11223344!!__ '

In [2]:

```
def double(str):
    return ''.join([c+c for c in str])
print(double('akash'))
```

```
aakkaasshh
```

In [3]:

```
print(double('String'))
```

```
SSttrriinngg
```

In [4]:

```
print(double('Hello World!'))
```

```
HHeelllloo  WWoorrlldd!!
```

In [5]:

```
print(double('1234!_ '))
```

```
11223344!!__
```

Question2

Create a function that reverses a boolean value and returns the string 'boolean expected'

if another variable type is given.

Examples

reverse(True) ➞ False

reverse(False) ➞ True

reverse(0) ➞ 'boolean expected'

reverse(None) ➞ 'boolean expected'

In [6]:

```
def reverse(arg=None):
    return not arg if type(arg) == bool else "boolean expected"

print(reverse(True)) # False
print(reverse(False)) # True
print(reverse(0)) # "boolean expected"
print(reverse(None)) # "boolean expected"
```

```
False
True
boolean expected
boolean expected
```

Question3

Create a function that returns the thickness (in meters) of a piece of paper after folding it n

number of times. The paper starts off with a thickness of 0.5mm.

Examples

num_layers(1) ➞ '0.001m'

# Paper folded once is 1mm (equal to 0.001m)

num_layers(4) ➞ '0.008m'

# Paper folded 4 times is 8mm (equal to 0.008m)

num_layers(21) ➞ '1048.576m'

# Paper folded 21 times is 1048576mm (equal to 1048.576m)

In [7]:

```
def num_layers(n):
    thickness = 0.5
    for _ in range(n):
        thickness *= 2

    return str(thickness / 1000)+'m' # for meters

print(num_layers(1))
print(num_layers(4))
print(num_layers(21))
```

```
0.001m
0.008m
1048.576m
```

Question4

Create a function that takes a single string as argument and returns an ordered list containing

the indices of all capital letters in the string.

Examples

index_of_caps('eDaBiT') ➞ [1, 3, 5]

index_of_caps('eQuINoX') ➞ [1, 3, 4, 6]

index_of_caps('determine') ➞ []

index_of_caps('STRIKE') ➞ [0, 1, 2, 3, 4, 5]

index_of_caps('sUn') ➞ [1]

In [8]:

```python
def index_of_caps(word):
    indices = []
    for i in range(len(word)):
        if word[i].isupper():
            indices.append(i)
    return indices

print(index_of_caps('BhaNu'))
print(index_of_caps('eDaBiT'))
print(index_of_caps('eQuINoX'))
print(index_of_caps('determine'))
print(index_of_caps('STRIKE'))
print(index_of_caps('sUn'))
```

```
[0, 3]
[1, 3, 5]
[1, 3, 4, 6]
[]
[0, 1, 2, 3, 4, 5]
[1]
```

Question5

Using list comprehensions, create a function that finds all even numbers from 1 to the given number.

Examples

find_even_nums(8) ➞ [2, 4, 6, 8]

find_even_nums(4) ➞ [2, 4]

find_even_nums(2) ➞ [2]

In [9]:

```
def find_even_nums(n):
    even =[x for x in range(2,n+1) if x % 2 == 0]
    return even

n = int(input('Enter a number : '))
find_even_nums(n)
```

```
Enter a number : 10
```

Out[9]:

```
[2, 4, 6, 8, 10]
```

In [10]:

```
find_even_nums(8)
```

Out[10]:

```
[2, 4, 6, 8]
```

In [11]:

```
find_even_nums(4)
```

Out[11]:

```
[2, 4]
```

In [12]:

```
find_even_nums(2)
```

Out[12]:

```
[2]
```