

Question1

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

Examples

```
double_char("String") ➔ "SSttrriinnngg"
double_char("Hello World!") ➔ "HHeellllloo Woorrlldd!!"
double_char("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"
```

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)
```

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]
```

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]
```