



## Question - 1

### Merge Strings

Complete the *mergeStrings* function in your editor. It has 2 parameters:

1. A string, *a*.
2. A string, *b*.

Your function must *merge* strings *a* and *b*, and then return a single merged string. A *merge* operation on two strings is described as follows:

- Append alternating characters from *a* and *b*, respectively, to some new string, *mergedString*.
- Once all of the characters in one of the strings have been merged, append the remaining characters in the other string to *mergedString*.

#### Input Format

The locked stub code in your editor reads two strings, *a* and *b*, from stdin and passes them to your function.

#### Constraints

- $1 \leq |a|, |b| \leq 25000$

#### Output Format

Your function must return the *merged* string. This will be printed to stdout by the locked stub code in your editor.

#### Sample Input 1

```
abc  
def
```

#### Sample Output 1

```
adbecf
```

#### Sample Input 2

```
ab  
zsd
```

#### Sample Output 2

```
azbsd
```

#### Explanation

*Sample Case 1*

*a* = *abc*

*b* = *def*

Taking alternate characters from both the strings , we get *adbecf*

*Sample Case 2*

*a* = *ab*

*b* = *zsd*

Taking alternate characters from both the strings , we get *azbsd*

## Question - 2

### Puzzle

---

We define the number of *holes* in an integer according to the following rules:

- 1, 2, 3, 5, and 7 each have *zero* holes.
- 0, 4, 6, and 9 each have *one* hole.
- 8 has *two* holes.

Complete the `countHoles` function in the editor below. It has one parameter: an integer, `num`. The function must return an integer denoting the total number of holes in `num`.

#### Input Format

Locked stub code in the editor reads a single integer, `num`, from stdin and passes it to the function.

#### Constraints

- $1 \leq num \leq 10^9$

#### Output Format

The function must return an integer denoting total number of holes in `num`. This is printed to stdout by locked stub code in the editor.

#### Sample Input 0

630

#### Sample Output 0

2

#### Explanation 0

The number 6 has *one* hole, the number 3 has *zero* holes, and the number 0 has *one* hole. Thus, we return the total number of holes in 630, which is  $1 + 0 + 1 = 2$ .

#### Sample Input 1

1288

#### Sample Output 1

4

#### Explanation 1

The number 1 has *zero* holes, the number 2 has *zero* holes, the number 8 has *two* holes, and the second number 8 also has *two* holes. Thus, we return the total number of holes in 1288, which is  $0 + 0 + 2 + 2 = 4$ .

## Question - 3

### Most frequent integer in an array

---

Find the most frequently occurring integer in an array of integers and return the number of times that it occurs.

If more than one integer occur for the maximum number of times, the program should still return that maximum number.

**Constraints** (where *numbers* is the input array and  $|numbers|$  is the array's length)

- $0 \leq |numbers| \leq 100$
- $-300000000 \leq numbers[i] \leq 300000000$ , where  $0 \leq i \leq |numbers|$

**Output Format**

Printing the contents of your returned integer to stdout is handled by the locked stub code in your editor.

**Sample Input**

*numbers* = [ 5, 5, 42 ]

**Sample Output**

*return* = 2

**Explanation**

The integer 5 appears twice in *numbers*, which is the maximum number of times that any specific integer appears in the array.