## Matrix Diagonal Sum

### Problem Statement:

Given a square matrix of size `N x N`, calculate the sum of the primary and secondary diagonals.

### Input Format:

- The first line contains an integer `N`, the size of the matrix.
- The next `N` lines each contain `N` integers separated by spaces, representing the matrix.

### Output Format:

Output two integers separated by a space:

- The sum of the primary diagonal.
- The sum of the secondary diagonal.

### Example:

#### Input:

```
3
1 2 3
4 5 6
7 8 9
```

#### Output:

```
15 15
```

#### Explanation:

- Primary diagonal: `1+5+9=15`
- Secondary diagonal: `3+5+7=15`

## Fizz Buzz with a Twist

### Problem Statement:

Print numbers from `1` to `N` with the following rules:

- If a number is divisible by `3`, print `"Fizz"`.
- If a number is divisible by `5`, print `"Buzz"`.

- If a number is divisible by both `3` and `5`, print `"FizzBuzz"`.
- If the number is a prime, print `"Prime"`.

**Input Format:**

A single integer `N`.

**Output Format:**

Output the required sequence, one item per line.

**Example:**

**Input:**

```
15
```

**Output:**

```
Prime
Prime
Fizz
Prime
Buzz
Fizz
Prime
Prime
Fizz
Buzz
Prime
Fizz
Prime
Prime
FizzBuzz
```

**Explanation:**

1. Numbers like 2,3,5,7,11,132, 3, 5, 7, 11, 132,3,5,7,11,13 are prime, so print "Prime".
2. Numbers divisible by 3 (but not 5) print "Fizz": 3,6,9,123, 6, 9, 123,6,9,12.
3. Numbers divisible by 5 (but not 3) print "Buzz": 5,105, 105,10.
4. Numbers divisible by both 3 and 5 print "FizzBuzz": 151515.

## Palindrome Checker

**Problem Statement:**

Determine whether a given string `S` is a palindrome. Ignore spaces, punctuation, and case.

### Input Format:

A single string `S`.

### Output Format:

Output `"YES"` if the string is a palindrome, otherwise output `"NO"`.

### Example:

**Input**:
A man, a plan, a canal, Panama
**Output**:
YES
**Explanation**:
Ignore spaces, punctuation, and case: "amanaplanacanalpanama".
Reverse of "amanaplanacanalpanama" is still "amanaplanacanalpanama", so the string is a palindrome.

## Prime Factorization

### Problem Statement:

Given a positive integer `X`, find all its prime factors.

### Input Format:

A single integer `X`.

### Output Format:

Print all prime factors of `X` in ascending order, separated by spaces. Each factor must appear as many times as it divides `X`.

### Example:

**Input:**

```
12
```

**Output:**

```
2 2 3
```

**Explanation:**

- Prime factors of 12: 12÷2=6, 6÷2=3, and 3 is a prime number.

## Password Validator

### Problem Statement:

You are given a string `S` representing a password. Determine if it is valid.
A password is valid if:

1. It is at least 8 characters long.
2. It contains at least one uppercase letter, one lowercase letter, and one digit.
3. It does not contain any spaces.

### Input Format:

A single string `S`.

### Output Format:

Output `"VALID"` if the password meets all criteria, otherwise output `"INVALID"`.

### Example:

#### Input:

```
Password123
```

#### Output:

```
VALID
```

#### Explanation:

- Length: 11 characters (valid
- Contains at least one uppercase letter: "P
- Contains at least one lowercase letter: "assword"
- Contains at least one digit: "123"
- No spaces: "Password123"

## Grade System Using List Comprehension

### Problem Statement:

You are given a list of scores, and you need to determine the corresponding grade for each score based on the following grading system:

- `A`: 90 and above
- `B`: 80-89
- `C`: 70-79
- `D`: 60-69
- `F`: below 60

Using **list comprehension**, generate a new list where each score is replaced by its corresponding grade.

**Input Format:** A list of integers, `scores []`, where each integer represents a score.

**Output Format:** A list of strings, where each string is the grade corresponding to the score in the input list.

### Example:

#### Input:

```
scores = [95, 67, 85, 78, 50, 91, 88, 72, 60]
```

#### Output:

```
['A', 'D', 'B', 'C', 'F', 'A', 'B', 'C', 'D']
```

#### Explanation:

- Score 95 → Grade `A` (because $95 \geq 90$)
- Score 67 → Grade `D` (because $60 \leq 67 < 70$)
- Score 85 → Grade `B` (because $80 \leq 85 < 90$)
- Score 78 → Grade `C` (because $70 \leq 78 < 80$)
- Score 50 → Grade `F` (because $50 < 60$)
- Score 91 → Grade `A` (because $90 \leq 91 < 100$)
- Score 88 → Grade `B` (because $80 \leq 88 < 90$)
- Score 72 → Grade `C` (because $70 \leq 72 < 80$)
- Score 60 → Grade `D` (because $60 \leq 60 < 70$)