Input: String

→ 1st # represents string
- range: negi billion – 1billion
- : indicates end of #

→ rest of string will contain chars
- 1 to 200 chars

Examples: [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]

1: some text → tpnf ufyu

-2:abc → yza

abc → "No # given for rotation"

-242 → "No string to encrypt"

:ed → "Not enough info"

42ed → "Not in correct format"

abc:4 → "Not in correct format"

Logic:

Parse the number
Parse the string
} if not in proper format then return

for each character:
if we rotate right:
- check if capital or lowercase
- Check if number
- perform rotation

else:
- check if capital or lowercase
- check if number
- perform rotation

Python

Suppose: uppercase = ['A', 'B', ..., 'Z']
lowercase = ['a', 'b', 'c', ..., 'z']
digits = [0, 1, ..., 9]

```
def caesarcipher(string):
    if not isinstance(string, str):
        return "Error. Incorrect Input"

    input = parseInput(string)
    rotation = parseRotation(string)
    if input == "" or rotation == MAXSIZE
        return "Error. Incorrect Input"

    # Loop through input and store it in a string
    encryption = ""
    for i in range(len(input)):
        char = input[i]
        index = getIndex(char)
        if letter.is_capital():
            encryption += rotate(index, uppercase, rotation)

        elif letter.is_lower():
            encryption += rotate(index, lowercase, rotation)

        else:
            #If not a letter, use digits array
            encryption += rotate(index, digits, rotation)
    return encryption
```

```
def rotate(index, list, rot):
    rot = rot % len(list)
    if rot >= 0:
        // rotate right
    else:
        // rotate left
    return
```

```
count = 0
while count != rot:
    if index == len(list) - 1:
        index = 0
    else: index += 1
    count += 1
return list[index]
```

```
count = 0
while count != rot:
    if index == 0
        index = len(list) - 1
    else: index -= 1
    count -= 1
return list[index]
```