# Gravity Falls Cryptogram Solver Version 3: Adding A1Z26 cipher

Help Dipper and Mabel discover the secret messages by writing a program that will help you uncover the mysteries of ciphers!

Now, you will add a third function, and your program will provide three potential solutions.

Dipper and Mabel will examine the output and choose the one that works best.

A1Z26 cipher is a substitution cipher. It means that it replaces the letter with its number in the alphabet.

For example:

- Letter **A** is the first letter in the alphabet, so it will be replaced with **1**.
- Letter **D** is the 4th letter in the alphabet, so it will be replaced with **4**.
- Thus, message **AD** is replaced with **1-4** (a dash is added between each two numbers) in the encrypted message..

Your program should correctly decrypt the numbers separated by dashes into corresponding uppercase letters. The dashes in the encrypted message **will be removed** from the decrypted message. All other characters (whitespace, punctuation, etc.) remain intact.

## Assumptions

1. You can safely assume that the original message will never contain any numbers or dashes.
2. The A1Z26 cipher is **case-insensitive**, so you will lose the information about the case of the letters. It's fine.

## What to do

In addition to the `decrypt_caesar` and `decrypt_atbash` functions that you already have, implement the `decrypt_a1z26` function that would decrypt a given message with the A1Z26 cipher.

1. Keep your `decrypt_caesar` and `decrypt_atbash` functions as is
2. Implement the `decrypt_a1z26` function that would decrypt a given message with the A1Z26 cipher
3. Alter your `main` function so that it now calls `decrypt_caesar, decrypt_atbash and decrypt_a1z26` functions and prints the outputs returned by the functions.

You **must** use the following template:

```python
# decrypt_caesar() function is already implemented, do not change it
# decrypt_atbash() function is already implemented, do not change it
def decrypt_a1z26(text: str) -> str:
    """
    Decipher a text (A1Z26 cipher).
    """
    # TODO: Implement this function
    pass

def main() -> None:
    """
    Main program.
    """
    text = input("Enter a text to decipher: ")
    print("Let's try all the methods we have:")
    # TODO: Alter the main() function, it should
    # call all three function and print
    # text deciphered with all three functions

main()
```

# Hints

- Consider converting the double digit numbers to letters before converting the single digit numbers (starting from 26 and working your way down to 1).
- Think how you can use the `range` function to generate a sequence of integers in descending order. Remember that some arguments of `range` can be negative (and it can take up to three arguments)!

- Consider using the `replace` method in the `str` class to implement the solution. Use `help(str.replace)` to find out more about this method.
- Consider using the replace method to replace the dashes with an empty string.

# Program name

Save your program as `gravity3.py`.

# Demo

https://asciinema.org/a/gk5XjvuqlHIDYMKeD6XC15Yqc

# Testing

To make sure your program works correctly, you should test it.

## Test Case 1

Run your program with `python gravity3.py`. Type `22-9-22-1-14 12-15-19 16-1-20-15-19 4-5 12-1 16-9-19-3-9-14-1.`, then press Enter. Your program should print:

```
Let's try all the methods we have:
Caesar cipher: 22-9-22-1-14 12-15-19 16-1-20-15-19 4-5 12-1
16-9-19-3-9-14-1.
Atbash cipher: 22-9-22-1-14 12-15-19 16-1-20-15-19 4-5 12-1
16-9-19-3-9-14-1.
A1Z26 cipher: VIVAN LOS PATOS DE LA PISCINA.
```

## Test Case 2

Run your program with `python gravity3.py`. Type `VWDQ LV QRW ZKDW KH VHHPV.`, then press Enter. Your program should print:

```
Let's try all the methods we have:
Caesar cipher: STAN IS NOT WHAT HE SEEMS.
```

```
Atbash cipher: EDWJ OE JID APWD PS ESSKE.
A1Z26 cipher: VWDQ LV QRW ZKDW KH VHHPV.
```

## Submitting

Submit `gravity3.py` via eClass.

*You may submit either all versions you complete, or only the final version.*