Gravity Falls Cryptogram Solver Version 2: Adding Atbash cipher

The mysteries of Gravity Falls must be discovered.

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 one more function to decrypt the message, and your program will provide two potential solutions (one with the Caesar cipher and one with the Atbash cipher).

Dipper and Mabel will choose the one that works best.

Atbash cipher reverses the alphabet.

For example:

- Letter **A** is the first letter in the alphabet, so it will be replaced with the last letter, which is **Z**.
- Letter **D** is the 4th letter in the alphabet, so it will be replaced with the 4th letter from the end, which is **W**.

Your program should correctly decrypt both upper-case and lower-case letters and keep all other characters (whitespace, punctuation, etc.) intact.

What to do

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

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

You **must** use the following template:

```python
# decrypt_caesar() function is already implemented, do not change it

def decrypt_atbash(text: str) -> str:
    """
    Decipher a text (Atbash 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 both functions and print
    # text deciphered with both functions

main()
```

Hints

- You might see that you use the same variable to store the alphabets in both functions. Maybe it's a good candidate for a global named constant.

Program name

Save your program as `gravity2.py`.

Demo

https://asciinema.org/a/oKA69E7riwyWSxPwtpGSkBPNz

Testing

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

Test Case 1

Run your program with `python gravity2.py`. Type `HLIIB, WRKKVI, YFG BLFI DVMWB RH RM ZMLGSVI XZHGOV.`, then press Enter. Your program should print:

```
Let's try all the methods we have:
Caesar cipher: EIFFY, TOHHSF, VCD YICF ASJTY OE OJ WJIDPSF UWEDLS.
Atbash cipher: SORRY, DIPPER, BUT YOUR WENDY IS IN ANOTHER CASTLE.
```

Run your program with `python gravity2.py`. Type `Sbwkrq lv ixq! :-)`, then press Enter. Your program should print:

```
Let's try all the methods we have:
Caesar cipher: Python is fun! :-)
Atbash cipher: Hydpij oe rcj! :-)
```

Submitting

Submit `gravity2.py` via eClass.

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