

MTH 3300 Problem Set 2

Due date: February 16, 2025 11:59pm

For the programming assignment, please follow the following naming convention for your Python files.
mth3300_<lastname>_<firstname>_pset2_part<#>.py

So for example, the name of my three files for this problem set would be:

- mth3300_abbariao_jaime_pset2_part1.py
- mth3300_abbariao_jaime_pset2_part2.py
- mth3300_abbariao_jaime_pset2_part3.py

If you worked with someone for this assignment, please note who on the PDF!

Unveiling a spy's real name (20 points)

Spies never go by their real name. Or do they?

- The first character of the real name is at index 2 of the encrypted name.
- The second character is at the second-to-last position of the encrypted name.
- The third character is at index 4 of the encrypted name.
- The fourth character is at the third-to-last position of the encrypted name.
- The fifth and final character is at the middle index of the encrypted name. Use floor division to find the middle index.

Write a program that takes a string of at least length 6 called `name`. If the name has a length less than 6, then we should print out that the name isn't associated with a spy! Otherwise, print out the spy's real name

Solution:

```
name = input("Enter a name: ")
n = len(name)
if n >= 6:
    real_name = f"{name[2]}{name[-2]}{name[4]}{name[-3]}{name[n // 2]}"
    print(real_name)
else:
    print(f"{name} is not associated with a spy!")
```

A simple encoder (20 points)

Write a program that takes a string of at least length 10 from the user. If the length of the string is less than 10, then we cannot encode this string. Otherwise, the rules for the string encoding are as follows:

- If the string has even length, then we'll swap the first half of the string with the second half.
- If the string has odd length, then we'll reverse the entire string
- Once either of the above conditions are met, we'll take the first two characters of the string and the last two characters and swap their positions! (Note that this requires that we create a new string)

Solution:

```
s = input("Enter a name: ")
n = len(s)

if n >= 10:
    if n % 2 == 0:
        mid = n // 2
        new_s = f"{s[mid:]}{s[:mid]}"
    else:
        new_s = s[::-1]

    print(f"{new_s[-2:]}{new_s[2:-2]}{new_s[:2]}")
else:
    print(f"Cannot encode this string")
```

Quadratic Formula (20 points)

Write a program that solves equations in the form of $ax^2 + bx + c = 0$.

- Prompt the users for the coefficients a , b , and c
- Calculate the discriminant $b^2 - 4ac$

- Determine the nature of the roots based on the discriminant
- Calculate and display the roots if they exist

Solution:

```
import math

a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))

discriminant = b ** 2 - 4 * a * c

if discriminant == 0:
    print(f"x = {-b / (2 * a)}")
elif discriminant > 0:
    print(f"x = {(-b + discriminant ** 0.5) / (2 * a)} and {(-b - discriminant ** 0.5) / (2 * a)}")
else:
    print("The roots are complex")
    # (optional)
    real = -b / (2 * a)
    imaginary = math.sqrt(-discriminant) / (2 * a)
    print(f"x = {real} + {imaginary}i and {real} - {imaginary}i")
```