**Name:**

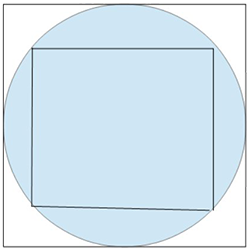
**Programming I**

**Lab Exercise 1.9.2025**

**Problem 1**

**A Circle and Two Squares**

Imagine a circle and two squares: a smaller and a bigger one. For the smaller one, the circle is a circumcircle and for the bigger one, an incircle.



Create a function, that takes an integer (radius of the circle) and returns the difference of the areas of the two squares.

**Examples**

square\_areas\_difference(5) ➞ 50

square\_areas\_difference(6) ➞ 72

square\_areas\_difference(7) ➞ 98

**Notes**

Use only positive integer parameters.

**Problem 2**

**Classes For Fetching Information on a Sports Player**

Create a class that takes the following four arguments for a particular football player:

* name
* age
* height
* weight

Also, create three functions for the class that returns the following strings:

* get\_age() returns "name is age age"
* get\_height() returns "name is heightcm"
* get\_weight() returns "name weighs weightkg"

### Examples

p1 = Player("David Jones", 25, 175, 75)

p1.get\_age() ➞ "David Jones is 25 years"

p1.get\_height() ➞ "David Jones is 175 cm"

p1.get\_weight() ➞ "David Jones weighs 75 kg"

#### Notes

name will be passed in as a string and age, height, weight will be integers.

**Problem 3**

**Two Distinct Elements**

In each input list, every number **repeats at least once**, except for **two**. Write a function that returns the **two unique numbers**.

### Examples

return\_unique([1, 9, 8, 8, 7, 6, 1, 6]) ➞ [9, 7]

return\_unique([5, 5, 2, 4, 4, 4, 9, 9, 9, 1]) ➞ [2, 1]

return\_unique([9, 5, 6, 8, 7, 7, 1, 1, 1, 1, 1, 9, 8]) ➞ [5, 6]

### Notes

Keep the same ordering in the output.

**Creating the Game Fermi**

Write a program that plays the game of Fermi. Generate three **distinct** random digits from 1 to 9. These digits are assigned to positions 1, 2, and 3. The goal of the game is for the player to guess the digits in the three positions in the least number of tries. For each guess, the player provides three digits for position 1, 2, and 3. The program replies with a hint consisting of Fermi, Pico, and Nano. If the digit guess for a given position is correct, then the reply is Fermi. If the digit guessed for a given position is in a different position, then the reply is Pico. If the digit guessed for a given position does not match any of the three digits, then the reply is Nano. Here is an example. If the three digits are 6, 5, and 8 at positions 1, 2, and 3 respectively.

|  |  |  |
| --- | --- | --- |
| Guess | Hint | Explanation |
| 1 2 5 | Nano Nano Pico | The value 5 matches at the wrong position |
| 8 5 3 | Pico Fermi Nano | The value 5 matches at the correct position. The 8 value matches but at the wrong position |
| 5 8 6 | Pico Pico Pico | All values match but at the wrong position |

Play games repeatedly until the player wants to quit. After each game, display the number of guesses made.

When your game is working, print out your documented source code and a sample printout of a game you played. Attach your source code and output to this sheet and turn in.