CSE 8A/6R Fall 2025 Homework 3

Due: Thursday, October 23, 2025 before 11:59pm

Full Name: Answer Key PID: 

Homework 3 topics covered:
1. For loops, 2. While loops

5 core competencies:
1. Code Tracing, 2. Code Comprehension, 3. Code Writing, 4. Debugging, and 5. Testing

Accessing printed worksheets: we typically have homework copies available during:

1. In-person office hours
2. Fridays 10 am - 2 pm in room B250 (where in-person study groups are happening)
3. Mondays 10 am - 6 pm in room B250 (where in-person study groups are happening)
4. Wednesdays 10 am - 6 pm in room B250 (where in-person labs are happening)
5. Tuesday lectures, in the lecture hall

Where B250 refers to the room on the CSE basement floor. We might have additional copies there outside of the above windows, but this is not guaranteed.

Submission

Questions 1 - 5:

- **Handwritten** in the given boxes only.
- Scanned/uploaded to Gradescope **Homework 3 Questions 1 - 5** assignment.

Question 6 (process journal):

- The **process journal** can be either handwritten or typed into a document.
- Uploaded to Gradescope **Homework 3 Question 6 Process Journal** assignment.

When you submit your work, please make sure to **assign the pages to the questions**. Otherwise, your work will also be ungradable.

Point breakdown for Homework 3:

- **Exemplary (3 homework points):** All problems receive a "Pass."
- **Satisfactory (2 homework points):** At least 4 out of 6 questions receive a "Pass."
- **Needs Improvement (1 homework point):** At least 1 out of 6 questions receive a "Pass."
- **Incomplete (0 homework points):** No problems receive a "Pass," or the assignment was not submitted, or the assignment is ungradable (e.g., illegible handwriting), or if the submission does not follow the instructions (e.g., homework is typed instead of handwritten, not answering in the boxes provided, not assigning pages to the question).

1. [Code Comprehension] In the following scenarios, which loop will you use and why?

A) You want to keep asking the user for a password until they enter the correct one.

Answer (circle one): For loop | While loop

Explanation:

We don't know how many attempts the user will take, so we use a while loop.

B) Check each letter in a word to see if it's a vowel.

Answer (circle one): For loop | While loop

Explanation:

We know how many letters are in a word, so we know how many times we need to run the loop for.

C) In a horror game, allow the user to walk down the hallway forever unless the user turns around and walks in the opposite direction.

Answer (circle one): For loop | While loop

Explanation:

We don't know when the user will change directions, so we should use a while loop.

2. [Code Tracing] What will be the output (printed) of running the following code? When showing your work, show how all values assigned to variables, including the variable `num`, change.

A)

```
sum = 0
for num in range(2, 8, 2):
    num_tripled = num * 3
    sum += num_tripled
print(sum)
```

Output:

36

Show your work:

num		2	4	6
num_tripled		6	12	18
sum	0	6	18	36

B)

```
word = "hello"
vowels = "aeiou"
for i in range(len(word)):
    if word[i] in vowels:
        to_print = word[i] * i
        print(to_print)
```

Output:

e

0000

Show your work:

i	0	1	2	3	4
word[i]	h	e	l	l	o
print		e			0000

C)

```
song_names = ["1989", "Reputation", "the 1", "Fearless", "Eras"]
for song in song_names:
   name_length = len(song)
   if name_length == 4:
       print(song)
```

Show your work:

song	"1989"	"Reputation"	"the 1"	"Fearless"	"Eras"
name_length	4	10	5	8	4
print?	✓	X	X	X	✓

Output:

1989
Eras

3. [Debugging] Oh no, infinite loops! For the following question, we wanted to print the numbers 5, 3 and 1, but we got into an infinite loop instead. Explain why this happens and how to fix it.

Note: if you encounter infinite loops, you can terminate the program by pressing `control + C` on your keyboard.

Code:	Expected output:	Actual output:

`number = 5`
`while number != 0:`
`  print(f"number = {number}")`
`  number = number - 2`
	number = 5
	number = 3
	number = 1
		number = 5
		number = 3
		number = 1
		number = -1
		number = -3
		… (keeps running!)

A) Explain: why did we enter an infinite loop?

The number variable will always be not equal to zero. As we keep subtracting by 2, number will reach 1 and then get to -1 in the next loop, skipping over zero.

B) Rewrite the corrected version of the above code in the box below. Your corrected code must still use a **while loop** and produce the expected output.

One possible solution:

```
number = 5
while number > 0:
    print(f"number = {number}")
    number = number - 2
```

4. Hailey is working in the study group on two questions:

Question A: print numbers 0, 1, 2

Question B: print numbers 0, 1, 2, 4, 5 (skipping 3)

Masha walked by and saw Hailey working on these two questions, and told Hailey that this question can be solved by changing just one line, using the knowledge they learned in class: `break` and `continue`.

NOTE: The following code is the same for both questions A and B. You have to find out what will you fill in for the placeholder "`# fill this line`" for two different scenarios shown below.

```
for i in range(6):
    if i == 3:
        # fill this line
    print(i)
```

Help Hailey fill in the code above using the following answer options and explain why it works!

Question A) Goal is to print numbers 0, 1, 2

Answer (circle one): `break` | `continue`

Explanation:

After i == 3, we no longer want to print out any more numbers, so we break out of the loop at this moment.

Question B) Goal is to print numbers 0, 1, 2, 4, 5 (skipping 3)

Answer (circle one): `break` | `continue`

Explanation:

To skip 3, we want to skip one iteration of the loop, so we use continue. This will skip the rest of the code in the loop and start the next iteration.

5. [Code Tracing, Code Comprehension] Consider the following code and an attempt at running the code

```
def guess_number():
    print("Please think of a number between 1 and 100 (both inclusive).")
    print("I will try to guess it!")
    # Set the lower and upper bounds
    low = 1
    high = 100

    while True:
        guess = (low + high) // 2
        print(f"Is your secret number {guess}?")

        feedback = input("h if guess is too high, l if too low, c if correct: ")

        if feedback == 'h':
            high = guess - 1  # Adjust the upper bound
        elif feedback == 'l':
            low = guess + 1   # Adjust the lower bound
        elif feedback == 'c':
            print(f"Great! I guessed your number: {guess}.")
            break

# Run the game
guess_number()
```

Output when the number in the user's mind is 12:

```
Please think of a number between 1 and 100 (inclusive).
I will try to guess it!
Is your secret number 50?
h if guess is too high, l if too low, c if correct: h
Is your secret number 25?
h if guess is too high, l if too low, c if correct: h
Is your secret number 12?
h if guess is too high, l if too low, c if correct: c
Great! I guessed your number: 12.
```

A) Show how the variables in the function `guess_number()` change when we run the code to produce the output above.

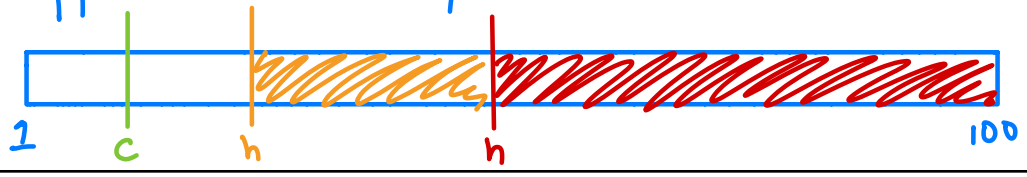
low	1	1	1
high	100	49	24
guess	50	25	12
feedback	h	h	c

B) How does the `guess_number` procedure stop running?

After the variable feedback is set to 'c' from user input, we would break out of the while loop and the procedure would stop running.

C) Explain in plain English (you can also draw, if that is helpful): how does this function guess a number the user is thinking about?

The function starts with a range of 1-100 and cuts the range in half depending on the user input. If the user input is 'h', then the larger half of the range is discarded. Then the opposite for an input of 'l'.



6. [Code Writing, Testing] Calculating the total amount of points the player earned in a game!

For this question, you are submitting a *process journal*. For detailed instructions, please check the instruction document: https://tinyurl.com/ProgrammingProcessJournal. Be sure to read this before you start implementing!

You are designing a 2D platformer game where the player can collect a gold coin (2 points), a silver coin (1 point), or fall into a trap and lose one point (-1 point). Design a program to calculate the total number of points that the player collected!

Name of the function: `calculate_points`
Parameter:

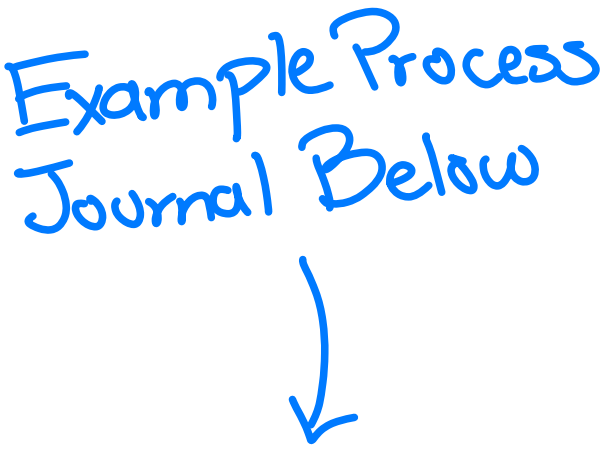
- `events`: *a list of strings*, representing the events that the player encountered.
  - `"gold"`: the gold coin, which should earn 2 points
  - `"silver"`: the silver coin, which should earn 1 point
  - `"trap"`: a trap, which should *reduce* 1 point
  - Any other string: ignore, 0 point.

Data type of return value: integer (int), representing the total amount of points (the value could be negative or positive).

Example of calling and testing the function:

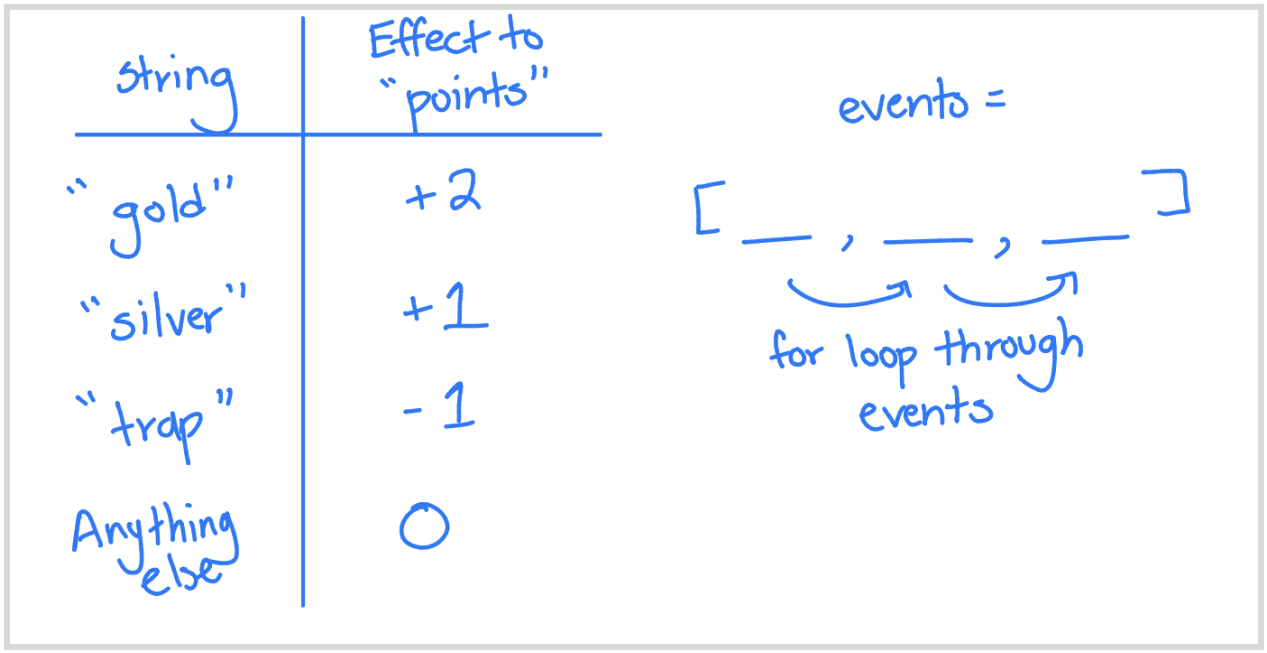
```
events_sample = ["gold", "silver", "trap", "mushroom", "trap"]
print(calculate_points(events_sample))  # Output should be 1
```

Explanation: points = 2 + 1 - 1 + 0 - 1 = 1



Example Process Journal for Homework 3

Understanding the Problem:
To understand the problem, I wrote out all of the possible conditions for the strings in events and how they would affect the points variable. Then I considered how I would loop through the events list, and I concluded that a for loop would be the best choice, as there is no need for an indefinite loop.



First version of code:
This was the first version of my code. I initialized a points variable and then looped through the events list. I checked if the events matched any of the words like “gold” and “silver”. I chose to use elif instead of another if because I know that for each loop, only one action can be executed. I excluded “trap” for the last condition and used else because I thought that there are only 3 possibilities that event could be.

```
def calculate_points(events):
    points = 0
    for event in events:
        if event == "gold":
            points += 2
        elif event == "silver":
            points += 1
        else:
            points -= 1
    return points
```

Running and Testing:
After running the code for the first time against the test case that was given in the handout, I realized that my code actually didn't work as expected. I assume it's because I used "else" instead of another "elif" in the code. I looked at the test case and saw that "mushroom" could also fall into the "else" case and subtract a point from points, so I knew this was the problem.

```python
def calculate_points(events):
    points = 0
    for event in events:
        if event == "gold":
            points += 2
        elif event == "silver":
            points += 1
        else:
            points -= 1
    return points


events_sample = ["gold", "silver", "trap", "mushroom", "trap"]
print(calculate_points(events_sample))  # Output should be 1
```

```
0
```

Debugging and Adjustments:
I changed the else statement to an elif statement to check if event was equal to “trap”. This fixed the issue and the test case passed!

```
def calculate_points(events):
    points = 0
    for event in events:
        if event == "gold":
            points += 2
        elif event == "silver":
            points += 1
        elif event == "trap":
            points -= 1
    return points


events_sample = ["gold", "silver", "trap", "mushroom", "trap"]
print(calculate_points(events_sample))  # Output should be 1
```

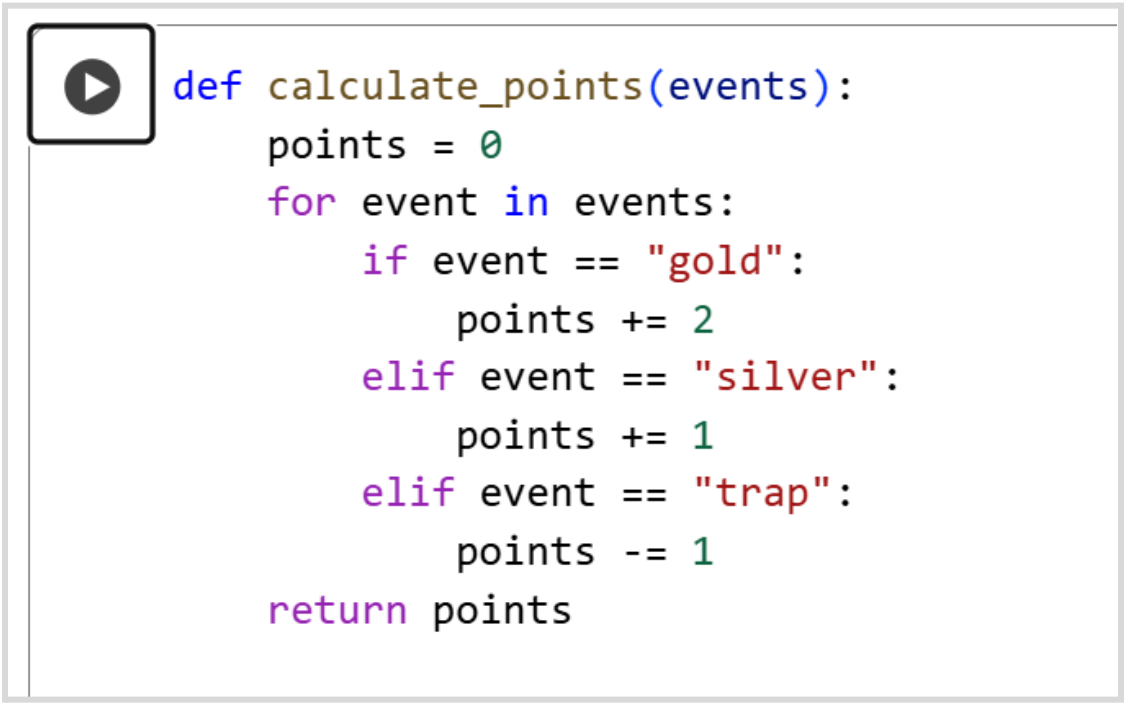
```
1
```

Running and Testing Again:
Some test cases I came up with are a list of 20 “gold”, an empty list, and a list of words that aren’t the three expected words. All of my tests passed!

```
events_sample = ["gold", "silver", "trap", "mushroom", "trap"]
print(calculate_points(events_sample))  # Output should be 1

test1 = ["gold"]*20
print("Test 1:", calculate_points(test1)) # Output should be 40

test2 = []
print("Test 2:", calculate_points(test2)) # Output should be 0

test3 = ["apple", "cse8a", "Silver"]
print("Test 3:", calculate_points(test3)) # Output should be 0
```

```
1
Test 2: 0
Test 3: 0
Test 1: 40
```

Final Version of My Code:



```
def calculate_points(events):
    points = 0
    for event in events:
        if event == "gold":
            points += 2
        elif event == "silver":
            points += 1
        elif event == "trap":
            points -= 1
    return points
```