

Name: _____

1. While Loop Tracing

For each of the following questions, determine the answer without running the code.

a. How many times does the following code print "JamCoders Rocks!"?

```
x = 0
while x < 10:
    print("JamCoders Rocks!")
    x += 1
```

b. How many times does the following code print "Woahohoh"?

```
x = 1

while x <= 50:
    print("Woahohoh")
    x *= 2
```

c. Does the following code cause an infinite loop?

```
answer = ""
while not False:
    answer = input("What is your favorite food?")
```

d. Does the following code cause an infinite loop?

```
num = 10
while num > 0:
    if num % 5 == 0:
        num += 1
    else:
        num -= 1
```

e. What value is printed?

```
x = 77
i = 0

while i < 7:
    x += 1
    i += 1

print(x)
```

f. What value is printed?

```
y = 4
i = 3

while i > 0:
    y *= i
    i -= 1

print(y)
```

2. Lost in Translation

Every for loop can be rewritten as a while loop. In this question, rewrite each for loop using a while loop instead.

a. Rewrite the following for loop using a while loop instead.

```
pets = ["cat", "dog", "goose", "chicken", "frog"]

for pet in pets:
    print(pet)
```

The equivalent while loop version:

```
pets = ["cat", "dog", "goose", "chicken", "frog"]

i = _____

while _____:
    pet = _____
    print(_____)
    _____
```

b. Rewrite the following for loop using a while loop instead.

```
total = 0

for num in range(-7, 37, 9):
    total += num
    print(num)

print("Total:", total)
```

The equivalent while loop version:

```
total = _____
num = _____

while _____:

    total += _____

    print(_____)

    num += _____

print("Total:", _____)
```

c. Rewrite the following for loop using a while loop instead.

```
for timer in range(10, -1, -1):
    print("Timer:", timer)

print("Blastoff!")
```

The equivalent while loop version:

```
timer = _____

while _____:

    print("Timer:", _____)

    _____

print(_____)
```

d. Rewrite the following for loop using a while loop instead.

```
food_list = ["patty", "patty", "krabby patty",
             "spongebob", "patrick", "krabby patty"]

for food in food_list:
    if food != "spongebob" and food != "patrick":
        print("yum! a", food)
    else:
        print("not a food!")
```

The equivalent while loop version:

```
food_list = ["patty", "patty", "krabby patty",
             "spongebob", "patrick", "krabby patty"]

i = _____

while _____:

    food = _____

    if _____ and _____:
        print("yum! a", food)
    else:
        print("not a food!")

    _____
```

3. Double Trouble

For each problem below, solve the problem twice: once using a for loop, and once using a while loop.

a. Write a program that prints every number from 0 to 7, inclusive. If the number is even, print that it is even. If the number is odd, print that it is odd.

For example, the output should look like this:

```
0 is even
1 is odd
2 is even
3 is odd
4 is even
5 is odd
6 is even
7 is odd
```

Implement the solution using a for loop:

```
for num in _____:
    if _____:
        print(_____)
    else:
        print(_____)
```

Now, implement the solution using a while loop:

```
num = _____
while _____:
    if _____:
        print(_____)
    else:
        print(_____)
    _____
```

b. Write a program that calculates the sum of all even numbers from 1 to 10, inclusive. At each step, print out the current number and the total sum being calculated.

For example, the output should look like this:

```
Current number: 2, total sum: 2
Current number: 4, total sum: 6
Current number: 6, total sum: 12
Current number: 8, total sum: 20
Current number: 10, total sum: 30
```

Implement the solution using a for loop:

```
total = _____

for num in _____:
    _____ += _____
    print("Current number:", _____, "total sum:", _____)
```

Now, implement the solution using a while loop:

```
total = _____
num = _____

while _____:
    _____ += _____
    print("Current number:", _____, "total sum:", _____)
    _____ += _____
```

c. You are given the alphabet as a reversed string. Print out all the letters in correct alphabetical order.

```
alphabet = "zyxwvutsrqponmlkjihgfedcba"
```

Implement the solution using a for loop:

```
for i in _____:
    print(_____)
```

Now, implement the solution using a while loop:

```
i = _____

while _____:
    print(_____)
    _____
```

4. Solo While Mission

All of the following exercises should be solved using a `while` loop, since we do not know ahead of time how many iterations we will need.

- a. Let's make a number guessing game that allows the user to guess the number until they get it right.

First, define a variable named `magic_number`.

magic_number = _____

Now, implement the `while` loop. On each iteration, ask for the user's input and check whether their guess is equal to `magic_number`.

Hint: You will need to cast the user's input from a string to an int using the `int()` function. For example, `int("35")` converts the string "35" to the integer 35.

[illegible]

b. Next, we'll make a Fortune Teller! The Fortune Teller will answer as many yes-or-no questions as we like. The program should keep asking for questions until the user enters the string "exit".

First, we'll define the list `fortune_answers` with all of the possible answers.

```
fortune_answers = [  
    "It is certain.",  
    "It is decidedly so.",  
    "Without a doubt.",  
    "Yes - definitely.",  
    "Most likely.",  
    "Outlook good.",  
    "Yes.",  
    "Signs point to yes.",  
    "Ask again later.",  
    "Don't count on it.",  
    "My reply is no.",  
    "Very doubtful."  
]
```

Next, create a variable called `question` to store the user's input. Initialize it with an empty string.

```
question = _____
```

Now, implement the `while` loop. The loop should continue until the user enters the string "exit".

For each iteration, ask for the user's yes-or-no question. If their question is not "exit", then generate a random index and use that index to access the `fortune_answers` list.

Important: Besides input and print, you may use the `randint` function. The `randint` function takes two numbers separated by a comma. For example, `randint(-3, 3)` generates a random number from -3 to 3, inclusive.

5. Counting

Define and implement the `countLongWords` function, which accepts a list of strings as its parameter and returns the number of words whose length is at least 5.

Examples:

- `countLongWords(["cat", "elephant", "dog", "zebra"])` returns 2
- `countLongWords(["apple", "pear", "banana"])` returns 2
- `countLongWords(["hi", "to", "be"])` returns 0

[illegible]