

Name: \_\_\_\_\_

1. (a) (20 points) Write a function `remove_e` that takes in a string and returns the same string with all of the e's removed. For example, running the following

```
example = "Let's go to the store."
```

```
print(remove_e(example))
```

should yield

```
Lt's go to th stor.
```

Write your function here:

- (b) (10 points) Rewrite `remove_e` to use only one line of code (in addition to the function declaration). Hint: use a string method. If your original function already did this, will automatically get these 5 points.

- (c) In the last part of this problem you will rewrite `remove_e` using recursion. Running
- ```
example = "Let's go to the store."  
print(remove_e(example))
```
- should still yield
- ```
Lt's go to th stor.
```
- (a) (3 points) What your base case? What should your function return in the base case?

- (b) (3 points) What is the general case? What should your function return in the general case?

- (c) (4 points) Write your recursive function here.

2. (30 points) The variable `scores` contains the score of each 2017 MSU football game. Complete the program below so that MSU's wins and losses are computed, regardless of the number of games played.

```
scores = [0, 31, 27, 31, 49, 21, 17, 25]
# [msu-score-1, opponent-score-1, msu-score-2, opponent-score-2, etc.]

# The missing code goes here but write it below.
# Assume that every game results in either a win or a loss.

print("MSU has", wins, "win(s) and", losses, "loss(es)")
```

Write the missing code here.

3. (30 points) Supply the missing function below. The missing function should draw the specified number of line segments, each with the specified segment length, starting at coordinate (0, 0) and moving to the right of the screen. The color of the sides should alternate between “blue” and “gold”.

```
import turtle

# The missing function goes here but write it below.

drawing_turtle = turtle.Turtle()
drawing_turtle.width(3)
drawing_turtle.hideturtle()

number_segments = int(input("Enter number of segments: "))
# Assume the user will enter an integer >= 1
segment_length = int(input("Enter length of a segment: "))
# Assume the user will enter an integer >= 10
bobcat_line(drawing_turtle, number_segments, segment_length)
```

Your function here: