1. You need to write code to do the following:

- Define the function $f(x) = 0.1x(1 - \frac{x}{100})$.
- Plot $f(x)$ in magenta for $x$ ranging between 0 and 150, with the axes labeled appropriately, of course.

Choose the correct two lines of code from the choices below to accomplish this.

**First line:**

A. $f(x) = 0.1x(1 - x/100)$
B. $f = 0.1x(1 - x/100)$
C. $f(x) = 0.1\times(1 - x/100)$
D. $f = 0.1\times(1 - x/100)$

**Second line:**

A. plot($f(x)$, 0, 150, color="magenta", axes_labels=["x", "f(x)"])
B. plot($f(x)$, (x, 0, 150), color="magenta", axes_labels=["x", "f(x)"])
C. plot($f(x)$, (x, 0, 150), color="magenta")
D. plot($f(x)$, (x, 0, 150), color="magenta", axes_labels=[x, f(x)])
E. plot($f(x)$, 0, 150, color="magenta", axes_labels=[x, f(x)])

**Solution:**

First line: C
Second line: B
2. The code below does not have any errors. If you run this code in SageMath, what will the output be?

```python
def squarefunc(x, y):
    a = x^2
    b = y^2
    return a + b

squarefunc(4, 7)
```

**Solution:** The code will print the number 65.
3. Your goal is to write code that will create a list of the numbers \( n^3 \), for values of \( n \) from 1 to 5 (inclusive). In other words, if after running your code, you type \texttt{print listofcubes} , the output would be \([1, 8, 27, 64, 125]\).

Choose the correct lines of code from the choices below to accomplish this.

**First line:**

A. \texttt{listofcubes = []}
B. \texttt{listofcubes = [1]}
C. \texttt{listofcubes = [1, 2, 3, 4, 5]}
D. \texttt{listofcubes = srange(1, 5)}
E. \texttt{listofcubes = srange(1, 6)}

**Second line:**

A. \texttt{for n in 1, 2, 3, 4, 5:}
B. \texttt{for n in [1, 2, 3, 4, 5]:}
C. \texttt{for n in srange(5):}
D. \texttt{for n in srange(1, 5):}
E. \texttt{for n in srange(1, 6):}

**Third line:**

A. \texttt{print n^3}
B. \texttt{n^3}
C. \texttt{listofcubes.append(n^3)}
D. \texttt{listofcubes + [n^3]}
E. \texttt{listofcubes.append[n^3]}

**Solution:**

First line: A  
Second line: B or E (they’re equivalent!)  
Third line: C
4. The goal of the following code is to define a function that returns the slope of the line through two points, with each point given as a list of x, y coordinates. For example the code 1 + slope([4,4], [2,19]) should output -13/2. However, the code below has five errors. Find all five of them, and explain how you would correct each one.

```
slope(pt1, pt2):
    x1 = pt1[0]
    y1 = pt1[1]
    x2 = pt2[0]
    y2 = pt2[1]
    s1o = (y2-y1)/(x2-x1)
    print slope
```

Solution:

```python
def slope(pt1, pt2):
    x1 = pt1[0]
    y1 = pt1[1]
    x2 = pt2[0]
    y2 = pt2[1]
    slope = (y2-y1)/(x2-x1)
    return slope
```
5. The goal of the code below is to iterate the function $f(x) = 2x$ five times with an initial value of 1. If the script worked correctly, it would display the output $[1, 2, 4, 8, 16, 32]$. However, it has five errors. Find all five of them, and explain how you would correct each one.

```python
mult2 = []
ums = srange(1,5)
for i in nums:
    test = 2*mult2[n]
    mult2.append(test)
mult # Should print [1, 2, 4, 8, 16, 32]
```

Solution:

```python
mult2 = [1]
ums = srange(0,5)
for n in nums:
    test = 2*mult2[n]
    mult2.append(test)
mult2 # Should print [1, 2, 4, 8, 16, 32]
```