CSCI 107, Second Practicum – Friday, March 10, 2021

Question One. 20 points. Use the python documentation to determine how to replace the three consecutive question marks with the function that calculates the kth binomial coefficient of a number n. As one example, if n is 10 and k is 5, the answer should be 252.

```
import math
n = int(input("Enter n: "))
k = int(input("Enter k: "))
print("The kth binomial coefficient of n =", math.???(n,k))
```

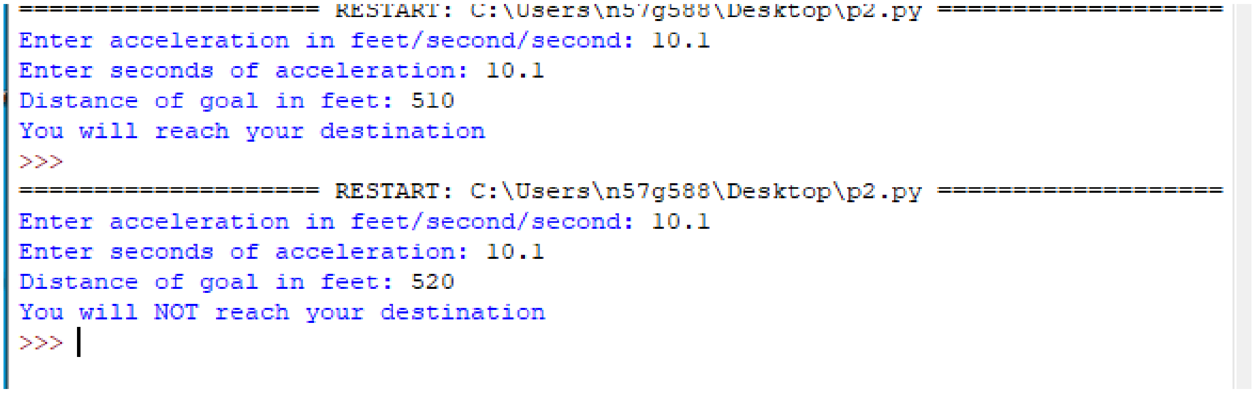
Question Two. 40 points. In the first practicum, we learned that
distance traveled = ½ * acceleration * time^2. Consider the following code:

```
acceleration = float(input("Enter acceleration in feet/second/second: "))
time = float(input("Enter seconds of acceleration: "))
goal = float(input("Distance of goal in feet: "))

if destination_reached(acceleration, time, goal):
    print("You will reach your destination")
else:
    print("You will NOT reach your destination")
```

Supply the missing function. The missing function should return True if the distance traveled is greater than or equal to the goal distance and False otherwise.

For example, 515.1505 feet will be covered if the acceleration is 10.1 feet/second/second and the time is 10.1 seconds. In the sample output below, a correct program is run twice. Once with a goal distance of 510 feet and once with a goal distance of 520 feet.



```
==================== RESTART: C:\Users\n57g588\Desktop\p2.py ====================
Enter acceleration in feet/second/second: 10.1
Enter seconds of acceleration: 10.1
Distance of goal in feet: 510
You will reach your destination
>>>
==================== RESTART: C:\Users\n57g588\Desktop\p2.py ====================
Enter acceleration in feet/second/second: 10.1
Enter seconds of acceleration: 10.1
Distance of goal in feet: 520
You will NOT reach your destination
>>>
```

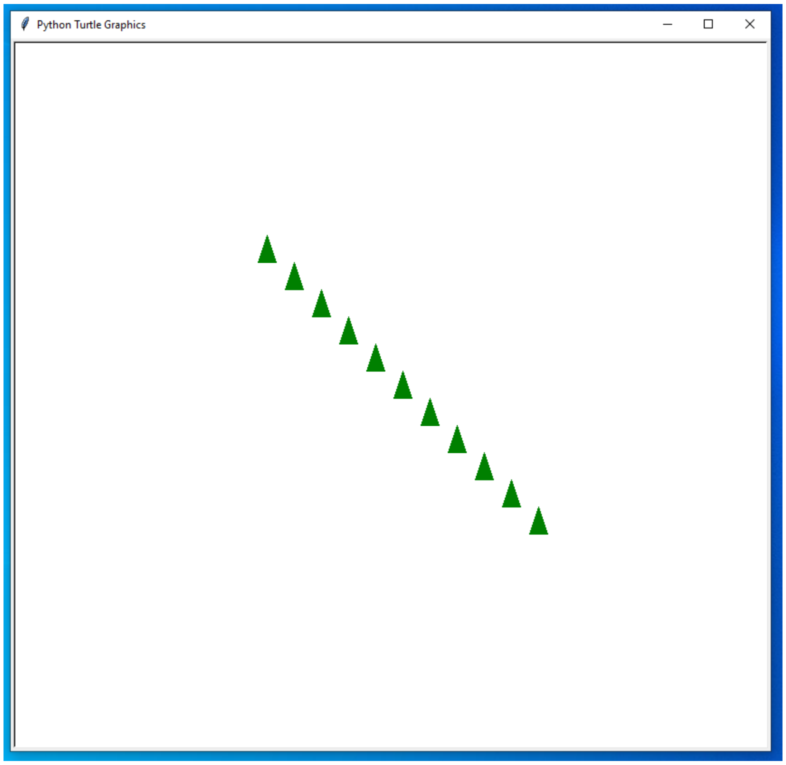
Question Three. 40 points. Supply the missing main() function below such that the drawing below is produced. The upper left tree is drawn at (-150, 150) and each subsequent tree is drawn 30 pixels to the right and 30 pixels down. For full credit, avoid repeating similar lines of code in your solution.

```
import turtle

def create_turtle():
    result = turtle.Turtle()
    result.color("green")
    result.hideturtle()
    return result

def draw_tree(tree, x, y):
    tree.penup()
    tree.goto(x, y)
    tree.pendown()
    tree.begin_fill()
    tree.goto(x+20, y)
    tree.goto(x+10, y+30)
    tree.goto(x, y)
    tree.end_fill()

main()
```


Python Turtle Graphics