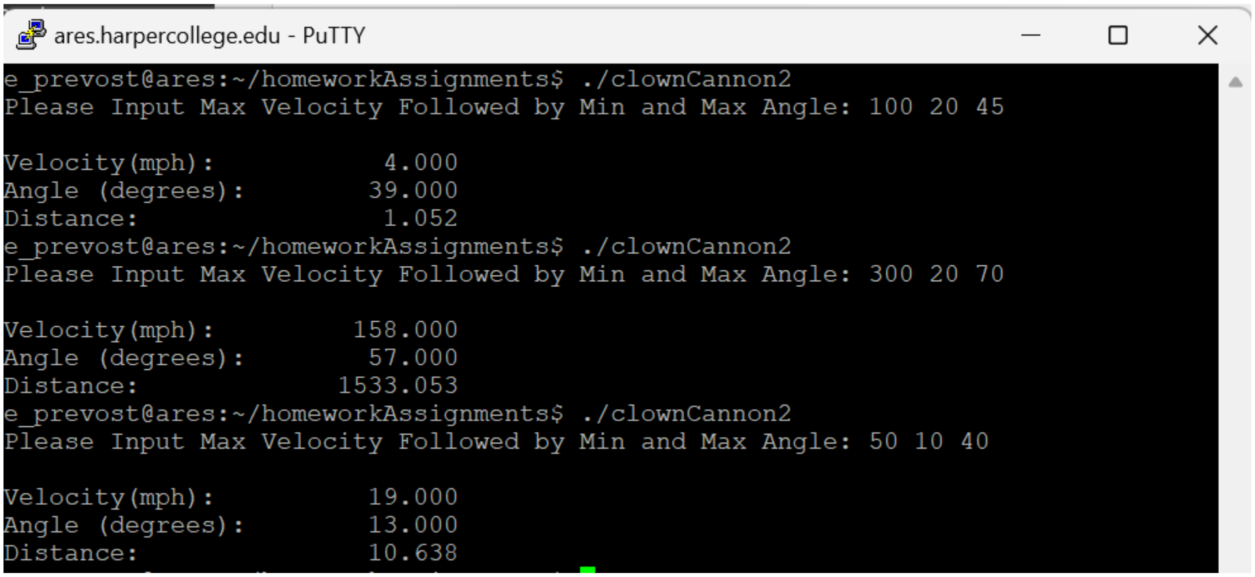
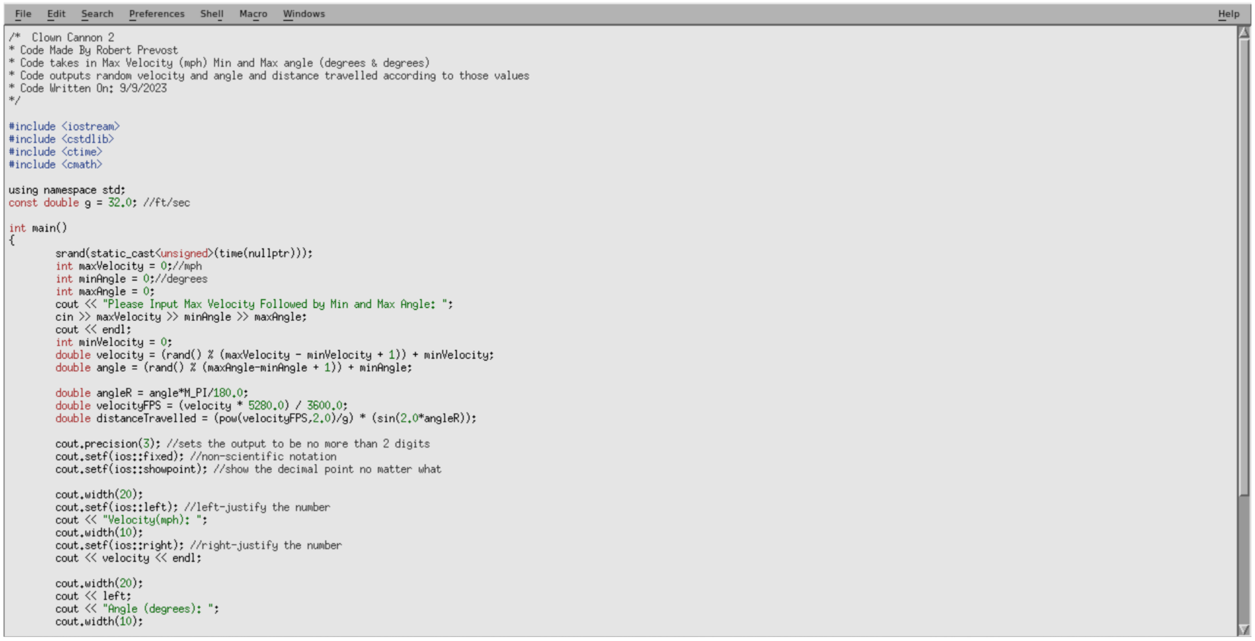
Clown Cannon 2 Assignment - Robert P.

Test Outputs:

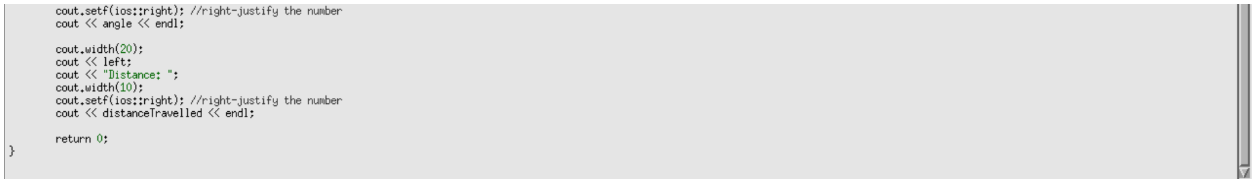


```
e_prevost@ares:~/homeworkAssignments$ ./clownCannon2
Please Input Max Velocity Followed by Min and Max Angle: 100 20 45

Velocity(mph):           4.000
Angle (degrees):        39.000
Distance:                1.052
e_prevost@ares:~/homeworkAssignments$ ./clownCannon2
Please Input Max Velocity Followed by Min and Max Angle: 300 20 70

Velocity(mph):         158.000
Angle (degrees):        57.000
Distance:             1533.053
e_prevost@ares:~/homeworkAssignments$ ./clownCannon2
Please Input Max Velocity Followed by Min and Max Angle: 50 10 40

Velocity(mph):          19.000
Angle (degrees):        13.000
Distance:               10.638
```

Source Code:



```
/*  Clown Cannon 2
* Code Made By Robert Prevost
* Code takes in Max Velocity (mph) Min and Max angle (degrees & degrees)
* Code outputs random velocity and angle and distance travelled according to those values
* Code Written On: 9/9/2023
*/

#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cmath>

using namespace std;
const double g = 32.0; //ft/sec

int main()
{
        srand(static_cast<unsigned>(time(nullptr)));
        int maxVelocity = 0;//mph
        int minAngle = 0;//degrees
        int maxAngle = 0;
        cout << "Please Input Max Velocity Followed by Min and Max Angle: ";
        cin >> maxVelocity >> minAngle >> maxAngle;
        cout << endl;
        int minVelocity = 0;
        double velocity = (rand() % (maxVelocity - minVelocity + 1)) + minVelocity;
        double angle = (rand() % (maxAngle-minAngle + 1)) + minAngle;

        double angleR = angle*M_PI/180.0;
        double velocityFPS = (velocity * 5280.0) / 3600.0;
        double distanceTravelled = (pow(velocityFPS,2.0)/g) * (sin(2.0*angleR));

        cout.precision(3); //sets the output to be no more than 2 digits
        cout.setf(ios::fixed); //non-scientific notation
        cout.setf(ios::showpoint); //show the decimal point no matter what

        cout.width(20);
        cout.setf(ios::left); //left-justify the number
        cout << "Velocity(mph): ";
        cout.width(10);
        cout.setf(ios::right); //right-justify the number
        cout << velocity << endl;

        cout.width(20);
        cout << left;
        cout << "Angle (degrees): ";
        cout.width(10);
```



```
        cout.setf(ios::right); //right-justify the number
        cout << angle << endl;

        cout.width(20);
        cout << left;
        cout << "Distance: ";
        cout.width(10);
        cout.setf(ios::right); //right-justify the number
        cout << distanceTravelled << endl;

        return 0;
}
```

Source Code Text:

```
/*  Clown Cannon 2
* Code Made By Robert Prevost
* Code takes in Max Velocity (mph) Min and Max angle (degrees & degrees)
* Code outputs random velocity and angle and distance travelled according to those values
* Code Written On: 9/9/2023
*/

#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cmath>

using namespace std;
const double g = 32.0; //ft/sec

int main()
{
        srand(static_cast<unsigned>(time(nullptr)));
        int maxVelocity = 0;//mph
        int minAngle = 0;//degrees
        int maxAngle = 0;
        cout << "Please Input Max Velocity Followed by Min and Max Angle: ";
        cin >> maxVelocity >> minAngle >> maxAngle;
        cout << endl;
        int minVelocity = 0;
        double velocity = (rand() % (maxVelocity - minVelocity + 1)) + minVelocity;
        double angle = (rand() % (maxAngle-minAngle + 1)) + minAngle;

        double angleR = angle*M_PI/180.0;
        double velocityFPS = (velocity * 5280.0) / 3600.0;
        double distanceTravelled = (pow(velocityFPS,2.0)/g) * (sin(2.0*angleR));

        cout.precision(3); //sets the output to be no more than 2 digits
        cout.setf(ios::fixed); //non-scientific notation
        cout.setf(ios::showpoint); //show the decimal point no matter what

        cout.width(20);
        cout.setf(ios::left); //left-justify the number
        cout << "Velocity(mph): ";
        cout.width(10);
        cout.setf(ios::right); //right-justify the number
        cout << velocity << endl;

        cout.width(20);
        cout << left;
        cout << "Angle (degrees): ";
        cout.width(10);
```

```
    cout.setf(ios::right); //right-justify the number
    cout << angle << endl;

    cout.width(20);
    cout << left;
    cout << "Distance: ";
    cout.width(10);
    cout.setf(ios::right); //right-justify the number
    cout << distanceTravelled << endl;

    return 0;
}
```