- **Name: yuyang tian**
- **NU ID: 002297971**

1 Cylindrical container

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main() {
    double radius;
    double height;
    double volume;
    double side;
    const double PI = 3.14159;
    cout << "input the radius of the base:" << endl;
    cin >> radius;
    cout << "input the height of a cylindrical container" << endl;
    cin >> height;

    volume = PI * radius * radius * height;
    side = cbrt(volume);
    cout << "the side of the cube with the same volume is " << fixed
<<setprecision(2)<< side << endl;
}
```

2 Plant tree in yard

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main() {
    double length;
    double radius;
    double space;
    const double PI = 3.14159;

    cout << "input the length of the yard:" << endl;
    cin >> length;
    cout << "input the radius of a fully grown tree:" << endl;
    cin >> radius;
    cout << "input the required space between fully grown trees:" << endl;
    cin >> space;

    double eachOccupied = 2*radius+space;
    int count = static_cast<int> (length/eachOccupied);
    double occupied = count * pow(radius,2.0) * PI;

    cout << fixed << setprecision(2);
    cout << "the number of trees that can be planted in the yard: " << count <<
endl;
    cout << "Total area occupied by the trees: " << occupied << endl;

    return 0;
}
```

3 Population growth

```
/*
The population of a town A is less than the population of town B.
However, the population of town A is growing faster than the population
of town B. Write a program that prompts the user to enter the population
and growth rate of each town. The program outputs after how many years
the population of town A will be greater than or equal to the population of
town B and the populations of both the towns at that time. (A sample input
is: Population of town A ¼ 5000, growth rate of town A ¼ 4%, population
of town B ¼ 8000, and growth rate of town B ¼ 2%.)
*/

#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double populationA, growthRateA, populationB, growthRateB;

    cout << "Enter the current population of town A: ";
    cin >> populationA;
    cout << "Enter the current population of town B: ";
    cin >> populationB;
    cout << "Enter the growth rate of town A: ";
    cin >> growthRateA;
    cout << "Enter the growth rate of town B : ";
    cin >> growthRateB;
    // Calculate the number of years required

    double years = ceil((log(populationB) - log(populationA)) / (log(1 + growthRateA/100) - log(1 + growthRateB/100)));
    cout << years << endl;
    // Calculate the number of years required
    for(int i = 0; i < years; i++) {
        populationA = floor(populationA*(1+growthRateA/100));
        populationB = floor(populationB*(1+growthRateB/100));
    }

    cout << "After " << years << " year(s) the population of town A will be greater than or equal to the population of town B" << endl;
    cout << "After " << years << " population of town A is " << populationA << endl;
    cout << "After " << years << " population of town B is " << populationB << endl;

    return 0;
}
```

4 Primes

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int primes[11] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31};
    string res;
    int num;
    bool isPrime = true;
    cout << "Number:" << endl;
    cin >> num;

    for (int prime : primes)
    {
        if(num % prime == 0) {
            isPrime = false;
            res += to_string(prime) + " ";
        }
    }
    if(isPrime) {
        cout << "Answers: Yes, it is a prime." << endl;
    } else {
        cout << "this number is not a prime, and it can be divided by " << res
<< "."<< endl;
    }
    return 0;
}
```

5 Novel income

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    double netPrice;
    int numberOfCopy;

    // Prompt the user to enter input
    cout << "Enter the Price: $";
    cin >> netPrice;
    cout << "Enter the copies: ";
    cin >> numberOfCopy;

    const double manuscriptRoyalty = 5000;
    const double publicationRoyalty = 20000;
    const double generalRate = 0.125;
    const double basicRate = 0.1;
    const double highRate = 0.14;

    double option1Royalty = manuscriptRoyalty + publicationRoyalty;
    double option2Royalty = numberOfCopy * netPrice * generalRate;
    double option3Royalty =
        (numberOfCopy > 4000) ? (4000 * netPrice * basicRate + (numberOfCopy -
4000) * netPrice * highRate) : (numberOfCopy * netPrice * basicRate);

    string bestOption = "O1";
    double highestRoyalty = option1Royalty;

    if (option2Royalty > highestRoyalty) {
        highestRoyalty = option2Royalty;
        bestOption = "O2";
    }

    if (option3Royalty > highestRoyalty) {
        bestOption = "O3";
    }
    cout << "Option 1: $" << option1Royalty << endl;
    cout << "Option 2: $" << option2Royalty << endl;
    cout << "Option 3: $" << option3Royalty << endl;
    cout << "Best Option: " << bestOption << endl;

    return 0;
}
```