附录一（问题一算法及结果）

```
#include<iostream>
using namespace std;
double max(double a, double b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
//向下取整
double quzheng(double a)
{
    if (a == int(a))  return int(a);
    else if (a<int(a + 1) && a>int(a + 1) - 1e-8)  return int(a + 1);
    else if (a > int(a) && a < int(a) + 1e-8) return int(a);
    else return int(a);
}
int doubleisint(double a, double b)
{
    if (fabs(a / b - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b - 1 - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b + 1 - (int)(a / b)) < 1e-8)
        return 1;
    else
        return 0;
}
int main()
{
    double x, y, z, u, w;
    cout << "问题一解法" << endl;
    for (x = 2; x < 5; x++)
    {
        double y1 = 1 / (1 - 1 / x);
        double y2 = 4 / (1 - 1 / x);
        for (y = max(quzheng(y1) + 1, x + 1); y < quzheng(y2); y++)
        {
            double z1 = 1 / (1 - 1 / x - 1 / y);
            double z2 = 3 / (1 - 1 / x - 1 / y);
            for (z = max(quzheng(z1) + 1, y + 1); z < quzheng(z2); z++)
            {
                double u1 = 1 / (1 - 1 / x - 1 / y - 1 / z);
                double u2 = 2 / (1 - 1 / x - 1 / y - 1 / z);
                for (u = max(quzheng(u1) + 1, z + 1); u <= quzheng(u2); u++)
                {
```

```
                    w = 1 / (1 - 1 / x - 1 / y - 1 / z - 1 / u);
                    if (doubleisint(w, x) && doubleisint(w, y) && doubleisint(w,
z) && doubleisint(w, u))
                    {
                        cout << "(" << x << " " << y << " " << z << " " << u <<"
" << w << ")" << endl;
                    }
                }
            }
        }
    }
    system("pause");
    return 0;
}
```

输出结果：

```
问题一解法
(2 3 7 43 1806) (2 3 7 44 924) (2 3 7 45 630) (2 3 7 48 336) (2 3 7 49 294) (2 3
7 56 168) (2 3 7 63 126)
(2 3 7 84 84) (2 3 8 25 600) (2 3 8 26 312) (2 3 8 27 216) (2 3 8 28 168) (2 3 8
30 120) (2 3 8 32 96)
(2 3 8 36 72) (2 3 8 48 48) (2 3 9 19 342) (2 3 9 20 180) (2 3 9 21 126) (2 3 9
24 72) (2 3 9 27 54)
(2 3 9 36 36) (2 3 10 16 240) (2 3 10 18 90) (2 3 10 20 60) (2 3 10 30 30) (2 3
12 13 156) (2 3 12 14 84)
(2 3 12 15 60) (2 3 12 16 48) (2 3 12 18 36) (2 3 12 24 24) (2 4 5 21 420) (2 4
5 22 220) (2 4 5 24 120)
(2 4 5 25 100) (2 4 5 30 60) (2 4 5 40 40) (2 4 6 13 156) (2 4 6 14 84) (2 4 6
15 60) (2 4 6 16 48)
(2 4 6 18 36) (2 4 6 24 24) (2 4 7 10 140) (2 4 7 14 28) (2 4 8 9 72) (2 4 8 10
40) (2 4 8 12 24)
(2 4 8 16 16) (2 5 6 8 120) (2 5 6 10 30)
```

附录二（问题二算法及结果）

```
#include<iostream>
using namespace std;
double max(double a, double b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
//向下取整
double quzheng(double a)
{
    if (a == int(a))  return int(a);
    else if (a<int(a + 1) && a>int(a + 1) - 1e-8)  return int(a + 1);
```

```
    else if (a > int(a) && a < int(a) + 1e-8) return int(a);
    else return int(a);
}
int doubleisint(double a, double b)
{
    if (fabs(a / b - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b - 1 - (int)(a / b)) < 1e-8)
        return 1;
    else if (fabs(a / b + 1 - (int)(a / b)) < 1e-8)
        return 1;
    else
        return 0;
}
int main()
{
    double x, y, z, w;
    cout << "问题二解法" << endl;
    for (x = 2; x < 5; x++)
    {
        double y1 = 1 / (1 - 1 / x);
        double y2 = 4 / (1 - 1 / x);
        for (y = max(quzheng(y1) + 1, x + 1); y < quzheng(y2); y++)
        {
            double z1 = 1 / (1 - 1 / x - 1 / y);
            double z2 = 3 / (1 - 1 / x - 1 / y);
            for (z = max(quzheng(z1) + 1, y + 1); z <= quzheng(z2); z++)
            {
                w = 2 / (1 - 1 / x - 1 / y - 1 / z);
                if (doubleisint(w, x) && doubleisint(w, y) && doubleisint(w, z))
                {
                    cout << "(" << x << " " << y << " " << z << " " << w << ")"
<< endl;
                }
            }
        }
    }
    system("pause");
    return 0;
}
```

输出结果：

```
问题二解法
(2 3 7 84)
(2 3 8 48)
(2 3 9 36)
(2 3 10 30)
(2 3 12 24)
(2 3 18 18)
(2 4 5 40)
(2 4 6 24)
(2 4 8 16)
(2 4 12 12)
(2 5 10 10)
```