int adaption(double s[][3], int row,int list);

using namespace std;

int main()

{

double s[13][3] = {

1, 2, 3.5,

2, 2.4 , 4.3,

3, 4, 5.4,

4, 2.7 , 5.3,

5, 3.9 , 5.35,

6, 2.9 , 7.1,

7, 3.2 , 7.2,

8, 2.22, 3.3,

9, 3.2 , 7.3,

10, 4, 9,

11, 5, 9.1,

12, 2.5 , 3.4,

13, 2.9 , 3.3,

};

int count = adaption(s, 13, 3);

for (int i = 0; i < 13 - count; i++)

{

for (int j = 0; j < 3; j++)

{

cout << s[i][j] << '\t';

}

cout << '\n';

}

}

int adaption(double s[][3], int row,int list)

{

int count = 0;//用来看合并了几次

for (int i = 0; i < row - 1; i++)

{

if (s[i][1] != 0)

{

for (int t = 1; t < row - 1 - i; t++)//看后面的可不可以合并,合并后就弄成0

{

if ((fabs(s[i][2] - s[i + t][2]) / ((s[i][2]> s[i + t][2])? s[i][2]: s[i + t][2])) <= 0.02)

{

s[i][1] = s[i][1] + s[i + t][1];

double K = (s[i][1] \* s[i][2] + s[i + t][1] \* s[i + t][2]) / (s[i][1] + s[i + t][1]);

s[i][2] = K;

for (int j = 1; j < list; j++)

{

s[i + t][j] = 0;

}

count++;

}

else

break;

}

}

}

//int jump = 0;//看要跳到哪个位置去；

for (int i = 0; i < row - 1; i++)

{

begin:if (s[i][1] == 0)//仅仅考虑到后面只有一行的情况。如果有多行空格怎么办

{

for (int j = i; j < row - 1; ++j)

{

s[j][0] = s[j + 1][0] - 1;

s[j][1] = s[j + 1][1];

s[j][2] = s[j + 1][2];

}

}

if (s[i][1] == 0)

goto begin;//如果多行为0 的话，再跳回去，重新执行

}

return count;

}