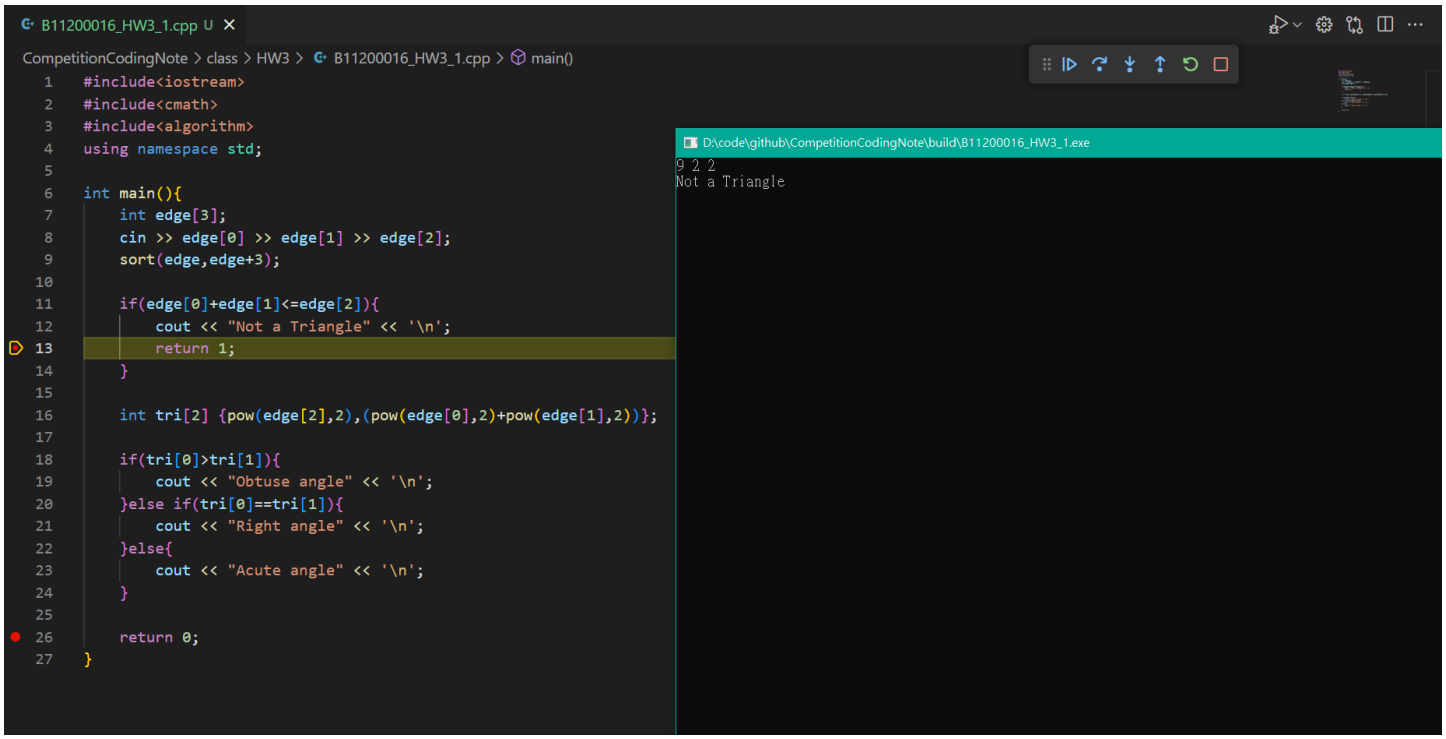
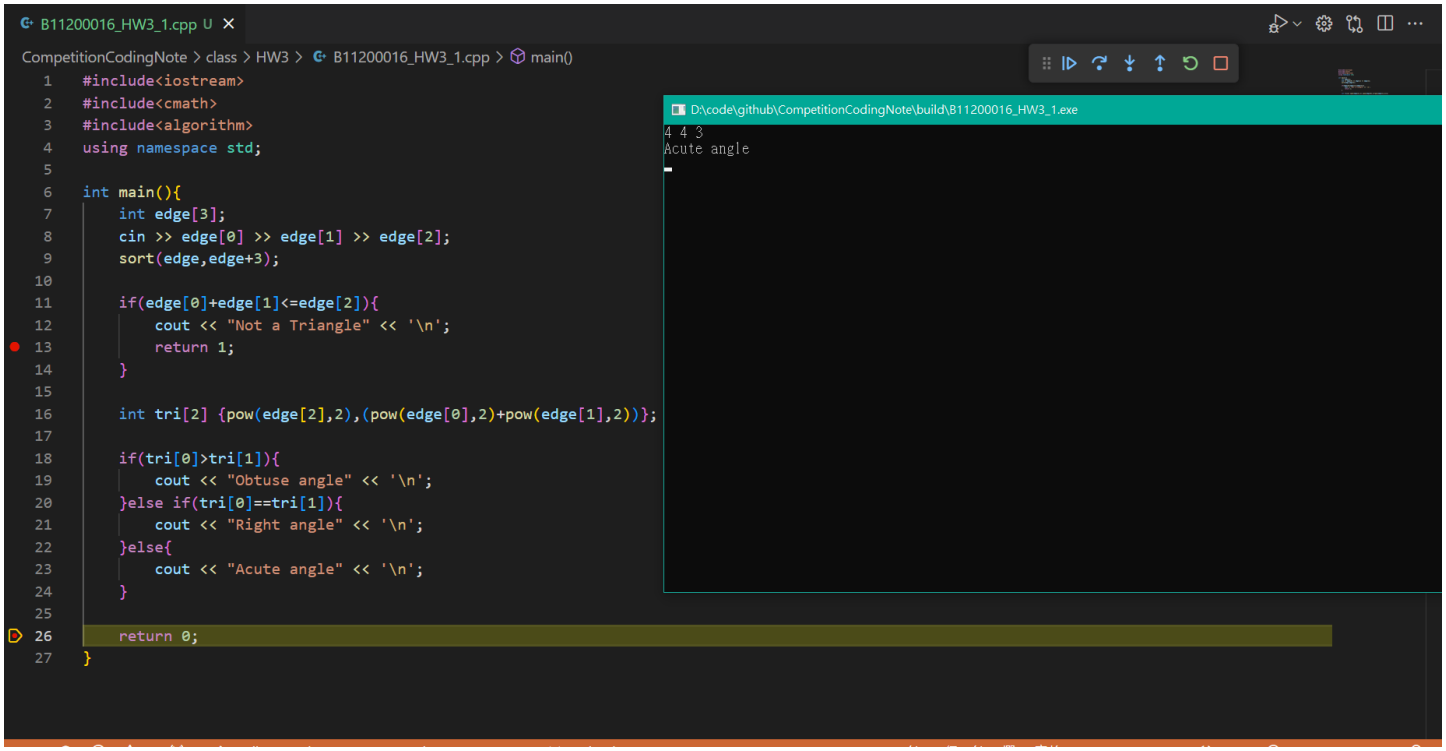
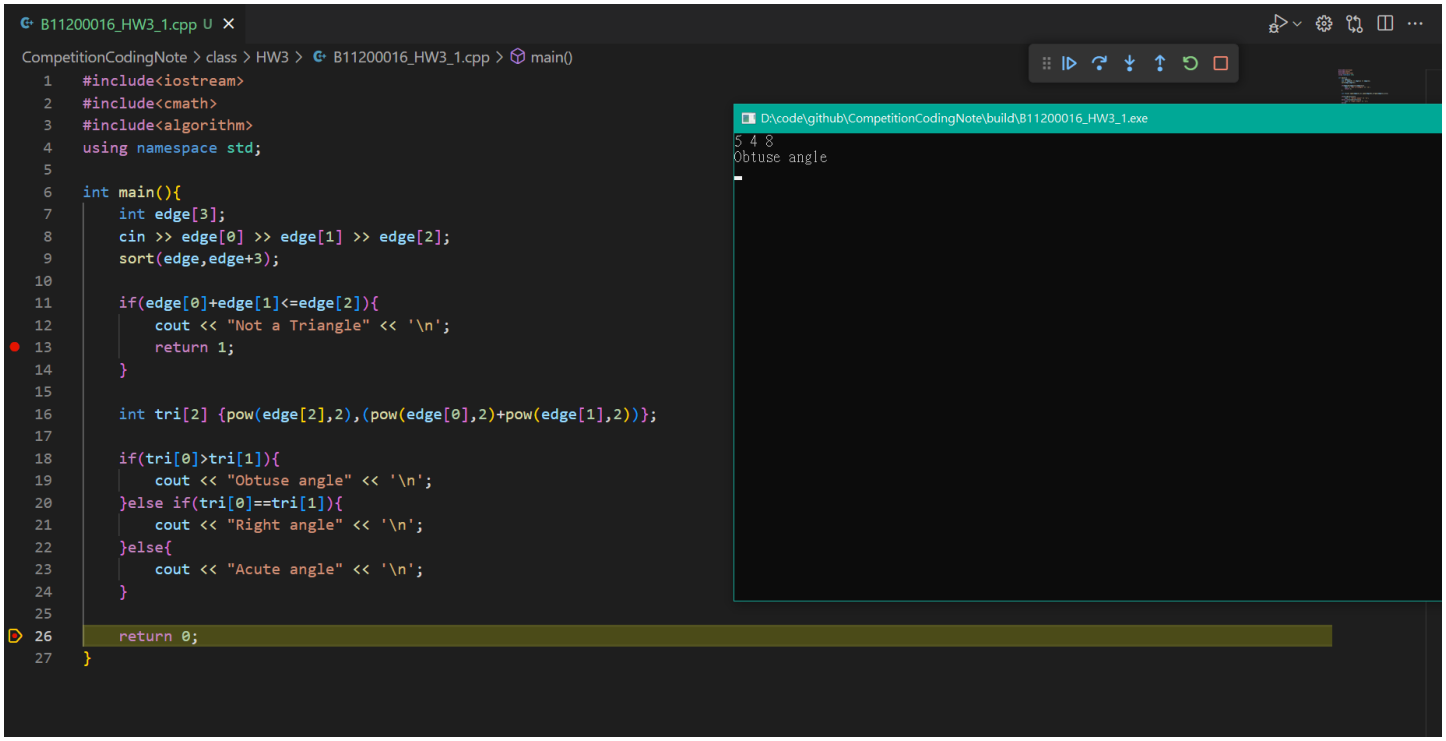
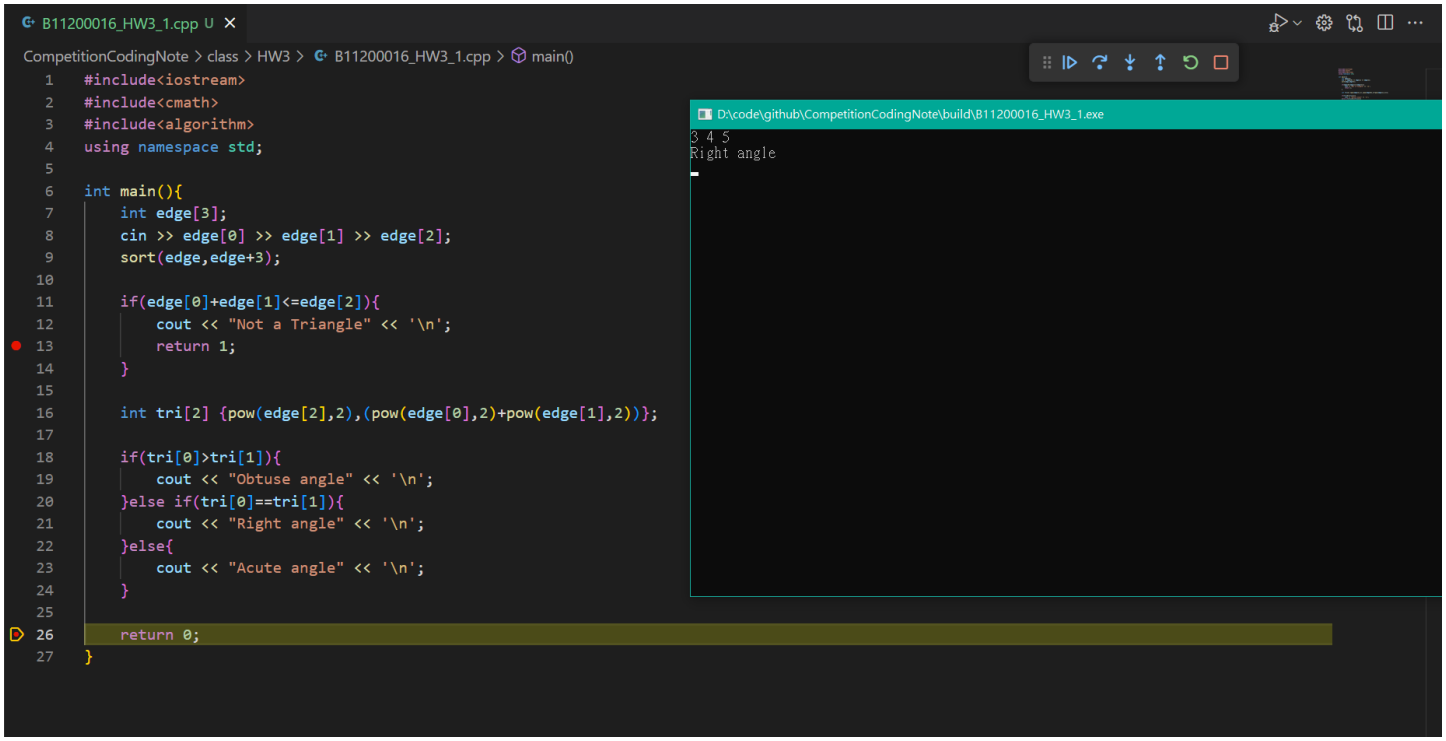
# 1.

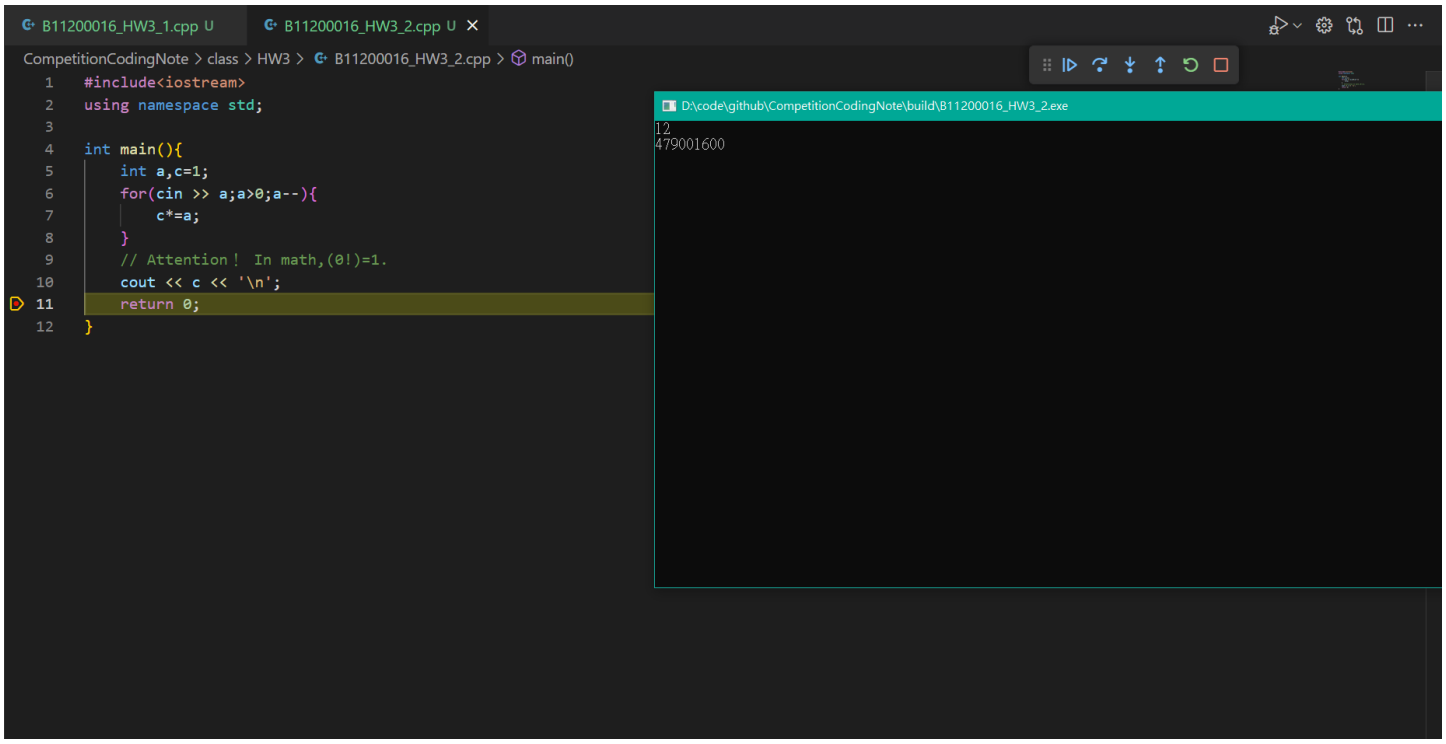

B11200016_HW3_1.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
#include<iostream>
#include<cmath>
#include<algorithm>
using namespace std;
int main(){
int edge[3];
cin >> edge[0] >> edge[1] >> edge[2];
sort(edge,edge+3);
if(edge[0]+edge[1]<=edge[2]){
cout << "Not a Triangle" << '\n';
return 1;
}
int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
if(tri[0]>tri[1]){
cout << "Obtuse angle" << '\n';
}else if(tri[0]==tri[1]){
cout << "Right angle" << '\n';
}else{
cout << "Acute angle" << '\n';
}
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
9 2 2
Not a Triangle

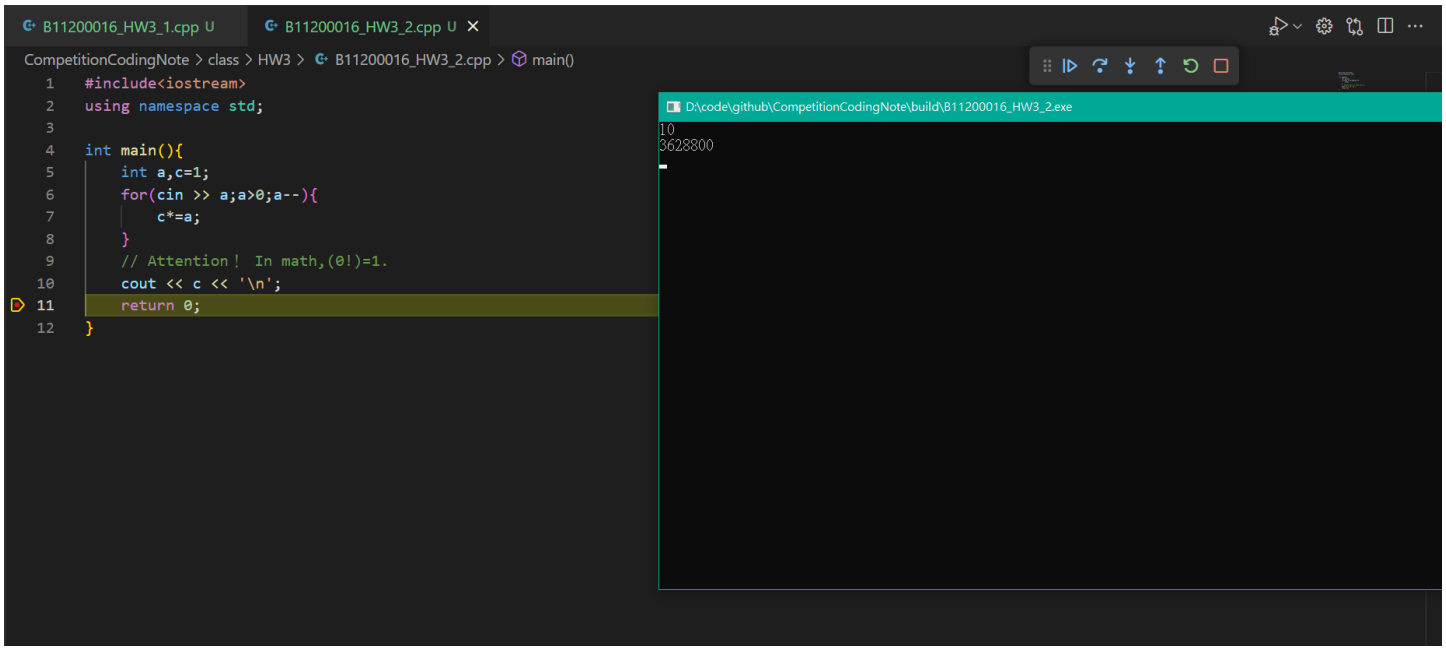

B11200016_HW3_1.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
#include<iostream>
#include<cmath>
#include<algorithm>
using namespace std;
int main(){
int edge[3];
cin >> edge[0] >> edge[1] >> edge[2];
sort(edge,edge+3);
if(edge[0]+edge[1]<=edge[2]){
cout << "Not a Triangle" << '\n';
return 1;
}
int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
if(tri[0]>tri[1]){
cout << "Obtuse angle" << '\n';
}else if(tri[0]==tri[1]){
cout << "Right angle" << '\n';
}else{
cout << "Acute angle" << '\n';
}
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
4 4 3
Acute angle

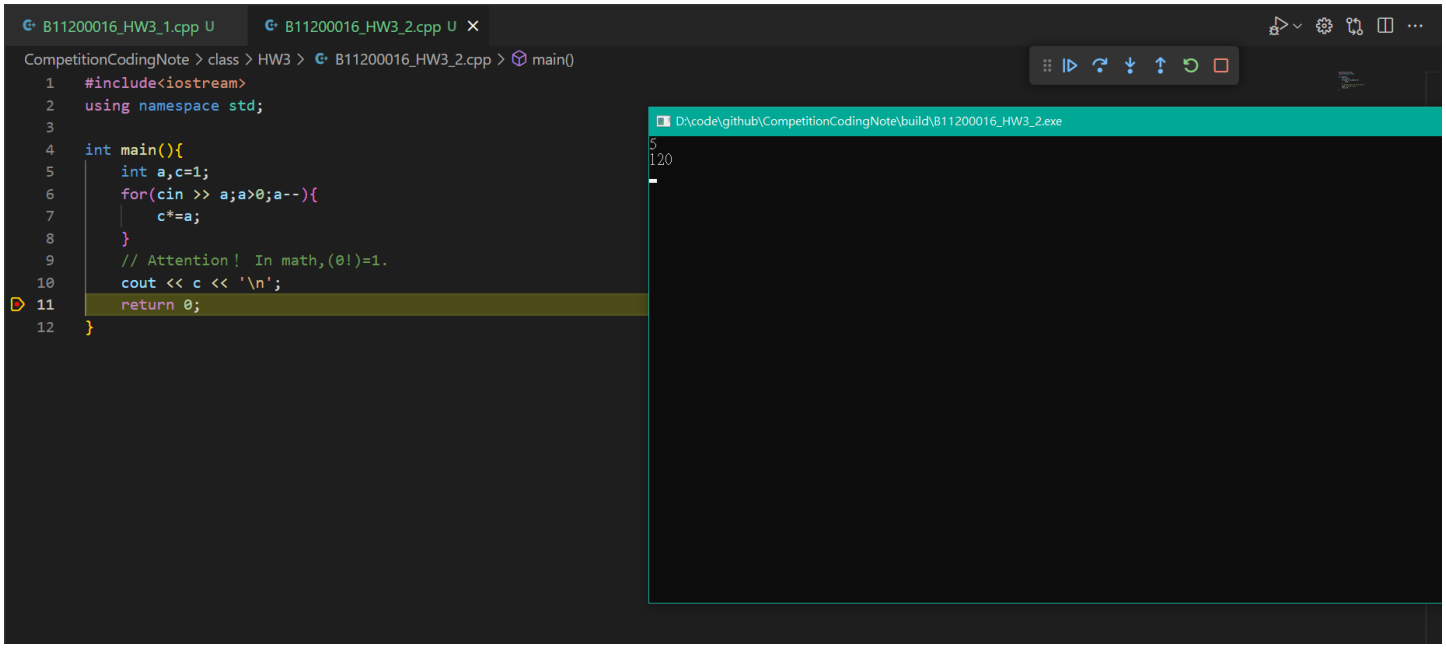

B11200016_HW3_1.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
#include<iostream>
#include<cmath>
#include<algorithm>
using namespace std;
int main(){
int edge[3];
cin >> edge[0] >> edge[1] >> edge[2];
sort(edge,edge+3);
if(edge[0]+edge[1]<=edge[2]){
cout << "Not a Triangle" << '\n';
return 1;
}
int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
if(tri[0]>tri[1]){
cout << "Obtuse angle" << '\n';
}else if(tri[0]==tri[1]){
cout << "Right angle" << '\n';
}else{
cout << "Acute angle" << '\n';
}
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
5 4 8
Obtuse angle


B11200016_HW3_1.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
#include<iostream>
#include<cmath>
#include<algorithm>
using namespace std;
int main(){
int edge[3];
cin >> edge[0] >> edge[1] >> edge[2];
sort(edge,edge+3);
if(edge[0]+edge[1]<=edge[2]){
cout << "Not a Triangle" << '\n';
return 1;
}
int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
if(tri[0]>tri[1]){
cout << "Obtuse angle" << '\n';
}else if(tri[0]==tri[1]){
cout << "Right angle" << '\n';
}else{
cout << "Acute angle" << '\n';
}
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
3 4 5
Right angle

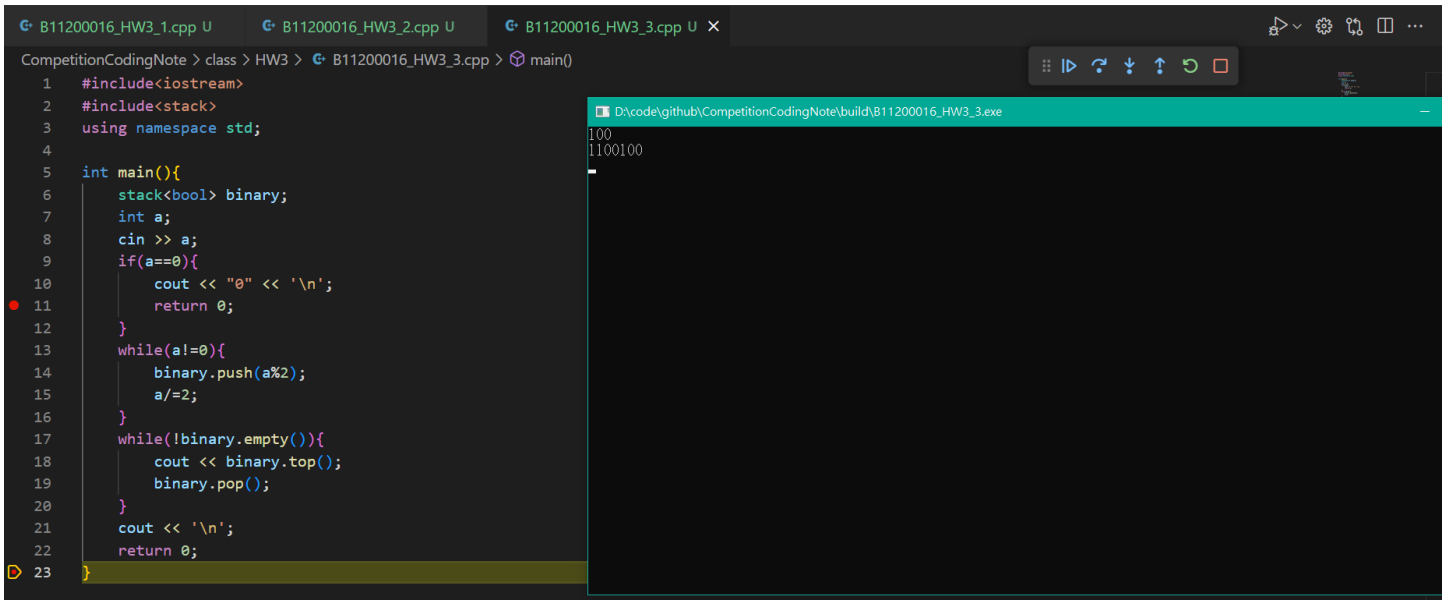
# 2.

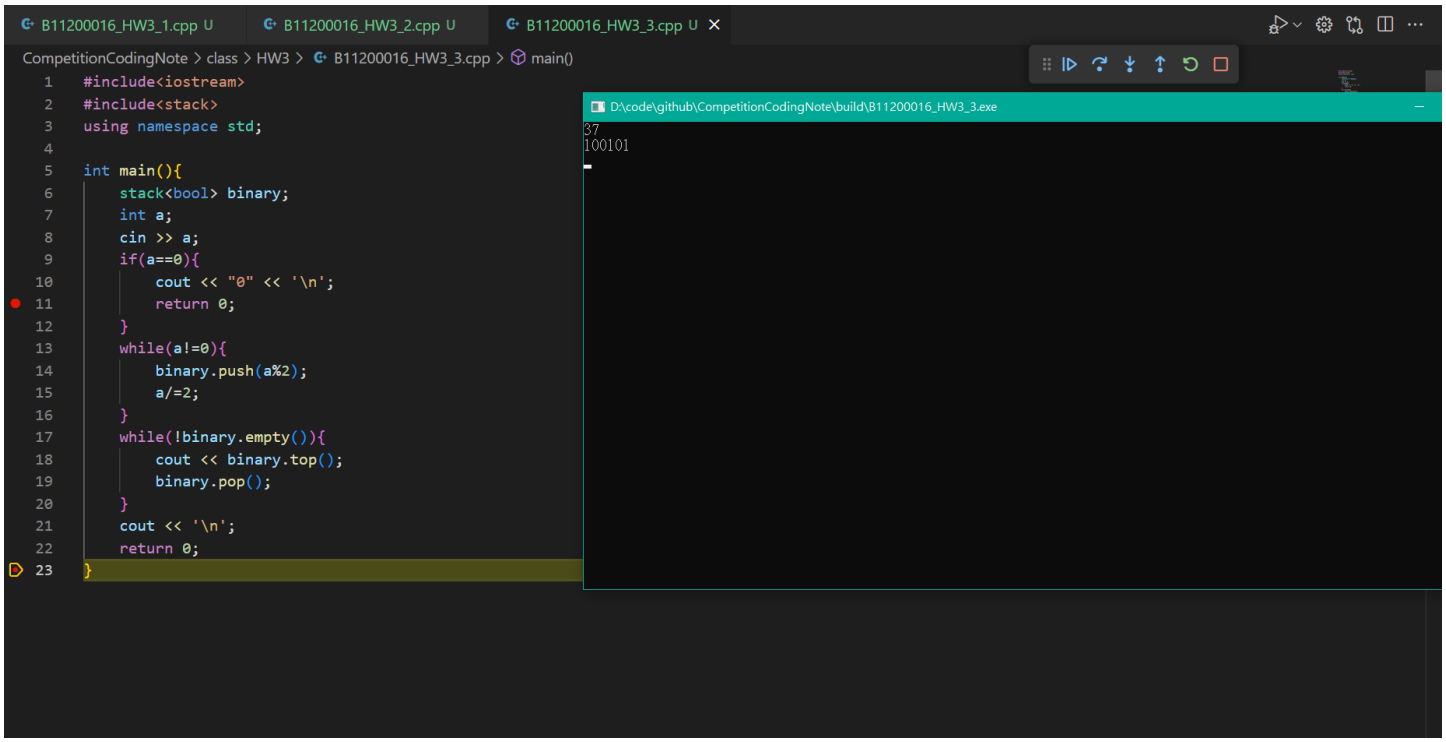

B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_2.cpp > main()
#include<iostream>
using namespace std;
int main(){
int a,c=1;
for(cin >> a;a>0;a--){
c*=a;
}
// Attention ! In math,(0!)=1.
cout << c << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe
12
479001600

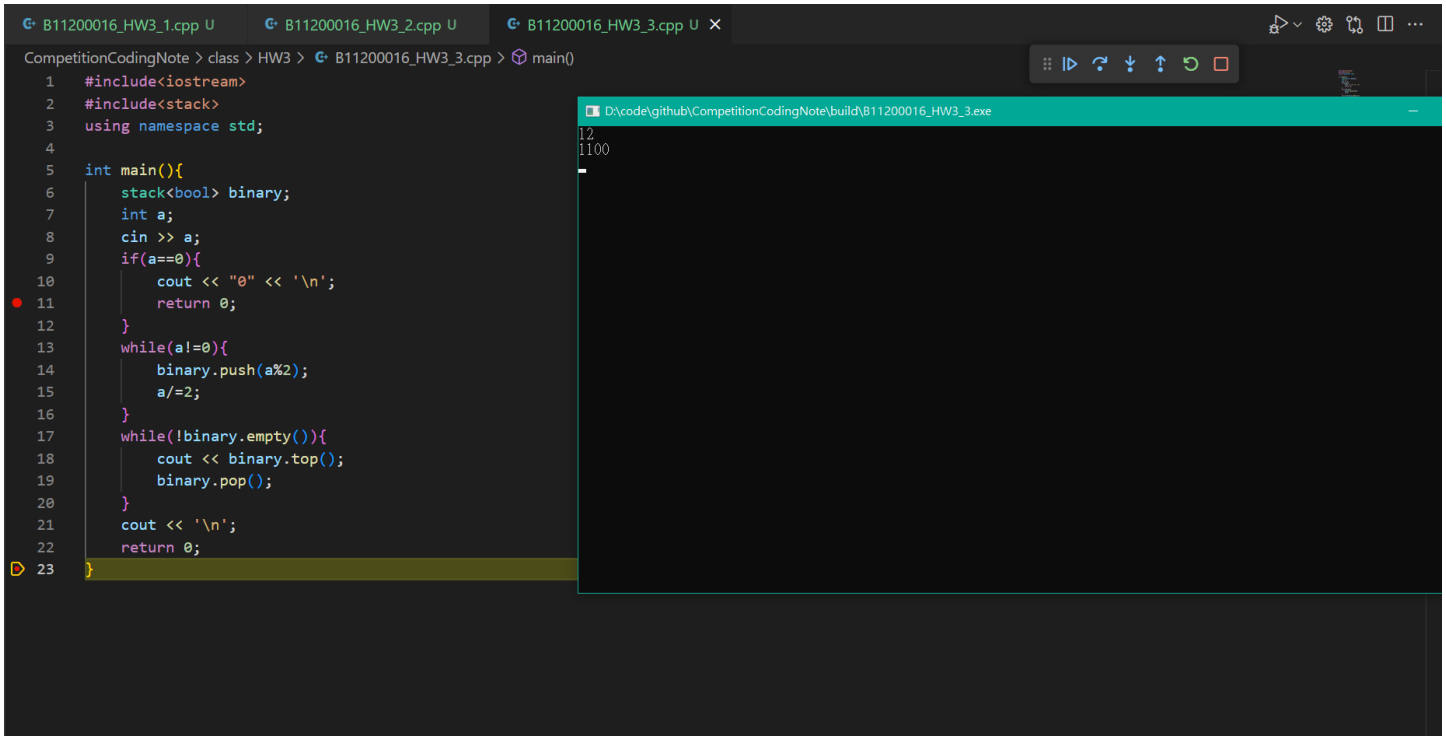

B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_2.cpp > main()
#include<iostream>
using namespace std;
int main(){
int a,c=1;
for(cin >> a;a>0;a--){
c*=a;
}
// Attention ! In math,(0!)=1.
cout << c << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe
10
3628800


B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_2.cpp > main()
#include<iostream>
using namespace std;
int main(){
int a,c=1;
for(cin >> a;a>0;a--){
c*=a;
}
// Attention ! In math,(0!)=1.
cout << c << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe
5
120

# 3.


B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
B11200016_HW3_3.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()
#include<iostream>
#include<stack>
using namespace std;
int main(){
stack<bool> binary;
int a;
cin >> a;
if(a==0){
cout << "0" << '\n';
return 0;
}
while(a!=0){
binary.push(a%2);
a/=2;
}
while(!binary.empty()){
cout << binary.top();
binary.pop();
}
cout << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_3.exe
100
1100100


B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
B11200016_HW3_3.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()
#include<iostream>
#include<stack>
using namespace std;
int main(){
stack<bool> binary;
int a;
cin >> a;
if(a==0){
cout << "0" << '\n';
return 0;
}
while(a!=0){
binary.push(a%2);
a/=2;
}
while(!binary.empty()){
cout << binary.top();
binary.pop();
}
cout << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_3.exe
37
100101


B11200016_HW3_1.cpp U
B11200016_HW3_2.cpp U
B11200016_HW3_3.cpp U
CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()
#include<iostream>
#include<stack>
using namespace std;
int main(){
stack<bool> binary;
int a;
cin >> a;
if(a==0){
cout << "0" << '\n';
return 0;
}
while(a!=0){
binary.push(a%2);
a/=2;
}
while(!binary.empty()){
cout << binary.top();
binary.pop();
}
cout << '\n';
return 0;
}
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_3.exe
12
1100