

1.

```
B11200016_HW3_1.cpp U X
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
1  #include<iostream>
2  #include<cmath>
3  #include<algorithm>
4  using namespace std;
5
6  int main(){
7      int edge[3];
8      cin >> edge[0] >> edge[1] >> edge[2];
9      sort(edge,edge+3);
10
11     if(edge[0]+edge[1]<=edge[2]){
12         cout << "Not a Triangle" << '\n';
13         return 1;
14     }
15
16     int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
17
18     if(tri[0]>tri[1]){
19         cout << "Obtuse angle" << '\n';
20     }else if(tri[0]==tri[1]){
21         cout << "Right angle" << '\n';
22     }else{
23         cout << "Acute angle" << '\n';
24     }
25
26     return 0;
27 }
```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
0 2 2
Not a Triangle

```
B11200016_HW3_1.cpp U X
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
1  #include<iostream>
2  #include<cmath>
3  #include<algorithm>
4  using namespace std;
5
6  int main(){
7      int edge[3];
8      cin >> edge[0] >> edge[1] >> edge[2];
9      sort(edge,edge+3);
10
11     if(edge[0]+edge[1]<=edge[2]){
12         cout << "Not a Triangle" << '\n';
13         return 1;
14     }
15
16     int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
17
18     if(tri[0]>tri[1]){
19         cout << "Obtuse angle" << '\n';
20     }else if(tri[0]==tri[1]){
21         cout << "Right angle" << '\n';
22     }else{
23         cout << "Acute angle" << '\n';
24     }
25
26     return 0;
27 }
```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe
4 4 3
Acute angle

```

B11200016_HW3_1.cpp U X
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
1  #include<iostream>
2  #include<cmath>
3  #include<algorithm>
4  using namespace std;
5
6  int main(){
7      int edge[3];
8      cin >> edge[0] >> edge[1] >> edge[2];
9      sort(edge,edge+3);
10
11      if(edge[0]+edge[1]<=edge[2]){
12          cout << "Not a Triangle" << '\n';
13          return 1;
14      }
15
16      int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
17
18      if(tri[0]>tri[1]){
19          cout << "Obtuse angle" << '\n';
20      }else if(tri[0]==tri[1]){
21          cout << "Right angle" << '\n';
22      }else{
23          cout << "Acute angle" << '\n';
24      }
25
26      return 0;
27  }

```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe

```

5 4 8
Obtuse angle

```

```

B11200016_HW3_1.cpp U X
CompetitionCodingNote > class > HW3 > B11200016_HW3_1.cpp > main()
1  #include<iostream>
2  #include<cmath>
3  #include<algorithm>
4  using namespace std;
5
6  int main(){
7      int edge[3];
8      cin >> edge[0] >> edge[1] >> edge[2];
9      sort(edge,edge+3);
10
11      if(edge[0]+edge[1]<=edge[2]){
12          cout << "Not a Triangle" << '\n';
13          return 1;
14      }
15
16      int tri[2] {pow(edge[2],2),(pow(edge[0],2)+pow(edge[1],2))};
17
18      if(tri[0]>tri[1]){
19          cout << "Obtuse angle" << '\n';
20      }else if(tri[0]==tri[1]){
21          cout << "Right angle" << '\n';
22      }else{
23          cout << "Acute angle" << '\n';
24      }
25
26      return 0;
27  }

```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_1.exe

```

3 4 5
Right angle

```

2.

```
CompetitionCodingNote > class > HW3 > B11200016_HW3_2.cpp > main()
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int a,c=1;
6      for(cin >> a;a>0;a--){
7          c*=a;
8      }
9      // Attention! In math,(0!)=1.
10     cout << c << '\n';
11     return 0;
12 }
```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe

12
479001600

```
CompetitionCodingNote > class > HW3 > B11200016_HW3_2.cpp > main()
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int a,c=1;
6      for(cin >> a;a>0;a--){
7          c*=a;
8      }
9      // Attention! In math,(0!)=1.
10     cout << c << '\n';
11     return 0;
12 }
```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe

10
3628800

The screenshot shows a C++ IDE with two tabs: `B11200016_HW3_1.cpp` and `B11200016_HW3_2.cpp`. The active tab is `B11200016_HW3_2.cpp`, which contains the following code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int a,c=1;
6     for(cin >> a;a>0;a--){
7         c*=a;
8     }
9     // Attention! In math, (0!)=1.
10    cout << c << '\n';
11    return 0;
12 }
```

The code is being executed, and the output window shows the result:

```
D:\code\github\CompetitionCodingNote\build\B11200016_HW3_2.exe
5
120
```

3.

```
CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()
1  #include<iostream>
2  #include<stack>
3  using namespace std;
4
5  int main(){
6      stack<bool> binary;
7      int a;
8      cin >> a;
9      if(a==0){
10         cout << "0" << '\n';
11         return 0;
12     }
13     while(a!=0){
14         binary.push(a%2);
15         a/=2;
16     }
17     while(!binary.empty()){
18         cout << binary.top();
19         binary.pop();
20     }
21     cout << '\n';
22     return 0;
23 }
```

D:\code\github\CompetitionCodingNote\build\B11200016_HW3_3.exe

```
100
1100100
```

```
CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()
1  #include<iostream>
2  #include<stack>
3  using namespace std;
4
5  int main(){
6      stack<bool> binary;
7      int a;
8      cin >> a;
9      if(a==0){
10         cout << "0" << '\n';
11         return 0;
12     }
13     while(a!=0){
14         binary.push(a%2);
15         a/=2;
16     }
17     while(!binary.empty()){
18         cout << binary.top();
19         binary.pop();
20     }
21     cout << '\n';
22     return 0;
23 }
```

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 x

CompetitionCodingNote > class > HW3 > B11200016_HW3_3.cpp > main()

12

1100

12

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

```
#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;
}
```