CPP Tracing 1

When working in a company, there are many programmers who work on the same program, and therefore each programmer has his own **part**, so it is necessary to learn **tracing** so that you can analyze the code and know what it will print without having to **run** the code, because you will not be able to collect the program code It's all run to find out the error, so you have to analyze your code, and find out if there are errors.

Here is an example:

```
#include <iostream>
#include <stdio.h>
using namespace std;

int main() {
        int x = 5;
        //define x by 5
            cout << x / 2 << endl;
        //result is going to be  2 because it's int
        int z = x * 100 / 2;
        //result is 5*100 / 2 = 250
            cout << z / 100 << "." << z % 100 << endl;
        //result 2.50
            cout << (float)x / 2 << endl;
```

```
        // we did casting so result in x is 2.50
        int y = 2;
        float a = x / y;
            cout << a << endl;
            a = 5.9;
            printf("%g\n", a);
        //%g means printing without 0 in the decimal part
        //result is going to be 5.9
        int b = 3111222333;
             printf("%d\n", b);
        /*
        The largest positive number that can be stored in an
        int is 2147483647
         so b can't be printing and it's going to be printed as
        negative number*/
        unsigned long long int c = 3111222333;
              printf("%llu \n", b);
        //now b is going to be printed, we used %llu
}
```

output:

```
2
2.50
2.5
2
```

5.9

-1183744963

3111222333