## input.txt

```
/* importing header */
import math;
import stdio;
import math;


// declaring function prototype
int add(int,int, float);
int scan();
void add(int, float, double);

/*
    multi-line comment
    /* support nested also */
*/


static void entryPoint(){

    // variable declaration
    int a;
    float b,c;

    println(a b c);
    // deleting variables
    discard a,b,c;

    /* variable initialization */
    int a = 10, b;
    int c = a <add> b, d = 10 <sub> a;
    println(a,b,c,d);

    double e = @max(2,3), f;
    println(e,f);

    discard c,d,e, /* valid comment */ f;

    // variable assignment
    a = 100;
    println("a: " a);

    b = a <rem> 11;
    println("b: " b);

    b = @max(a,101);
    println("b: " b);
    discard a,b;

    /* condition */
    float a = 10;
    float b = 100;

    justInCase(a <lt> b){
        println("Actually executed since " a "< " b);

        justInCase(10 <lt> 100){
            println("Also executed");
        }

        justInCase(10 <gt> b){
            println("Not executed since false");
```

```
        }
    }

    println("------------- separator -----------------");

    a = 10; int c = 15;
    justInCase(5 <lt> 10 and a <lt> c){
        println("5 < 10 and " a " < " c);
    }

    justInCase(5 <lt> 10 and a <lt> b and 10 <gt> 10 ){
        println("Condition is false");
    }

    int i = 0;
    till( i <lt> 100){
        i = i <add> 15;
    }

    /* library function */

    int n = @scanInt();
    println(n);

    float f = @scan();
    println("Value after scan is: " f);

    discard n,f;

    int m = 6, n = 4;
    int result = m <mul> n;
    @show(result);

    double mx = @max(2.5,5);
    @show(mx);

    float rootf = @sqrt(42);
    println("Square root of 42 is :" rootf "(float)");

    int toFind = -42;
    n = @sqrt(-42);
    println("Square root of " toFind " is :" n "(int)");

    n = @toInt(rootf);
    println(rootf " becomes " n " after toInt" );

    rootf = @toFloat(n);
    println(n " becomes " rootf " after toFloat" );

    rootf = @toDouble(rootf);
    println(rootf " becomes " rootf " after toDouble" );

    // user defined function
    int n1, n2;
    float f1;

    int addRes = @add(n1,n1,f1);
    println(addRes);

    @add(m,f1,100);

    int dum = @add(m,f1,100);

    @test();
}
```