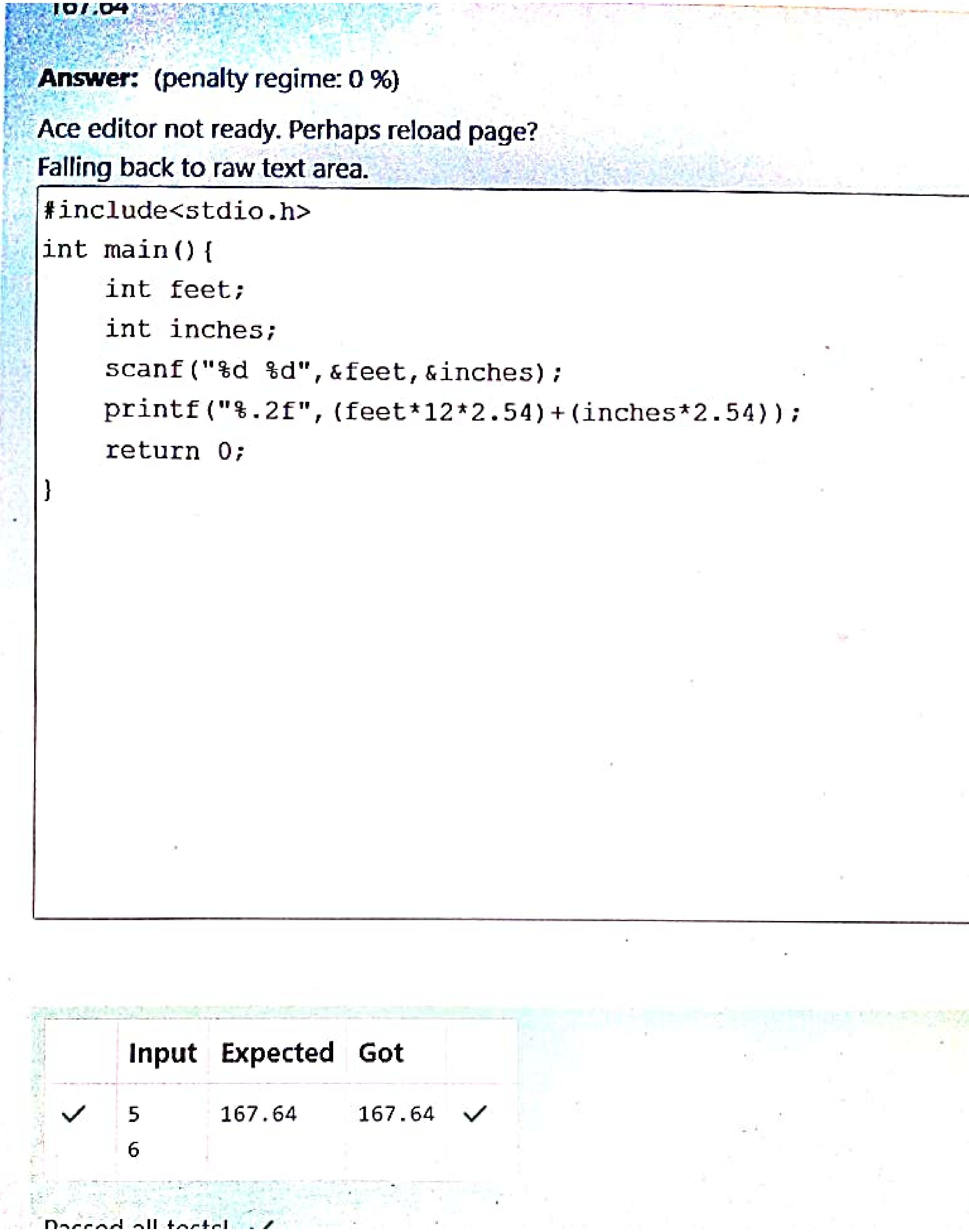


167.64

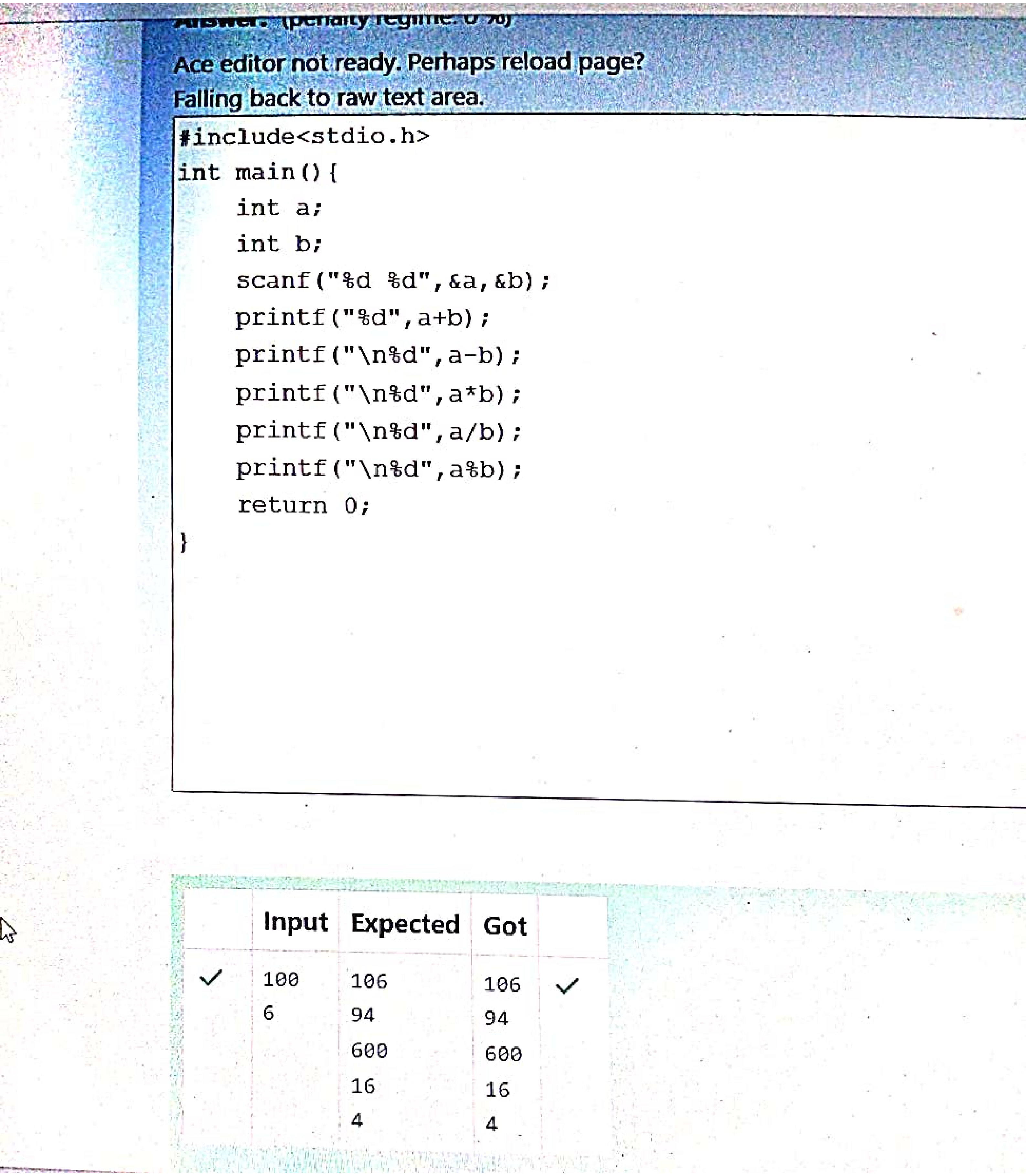
Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
#include<stdio.h>
int main(){
    int feet;
    int inches;
    scanf("%d %d",&feet,&inches);
    printf("%.2f",(feet*12*2.54)+(inches*2.54));
    return 0;
}
```

	Input	Expected	Got	
✓	5 6	167.64	167.64	✓

Passed all tests! ✓

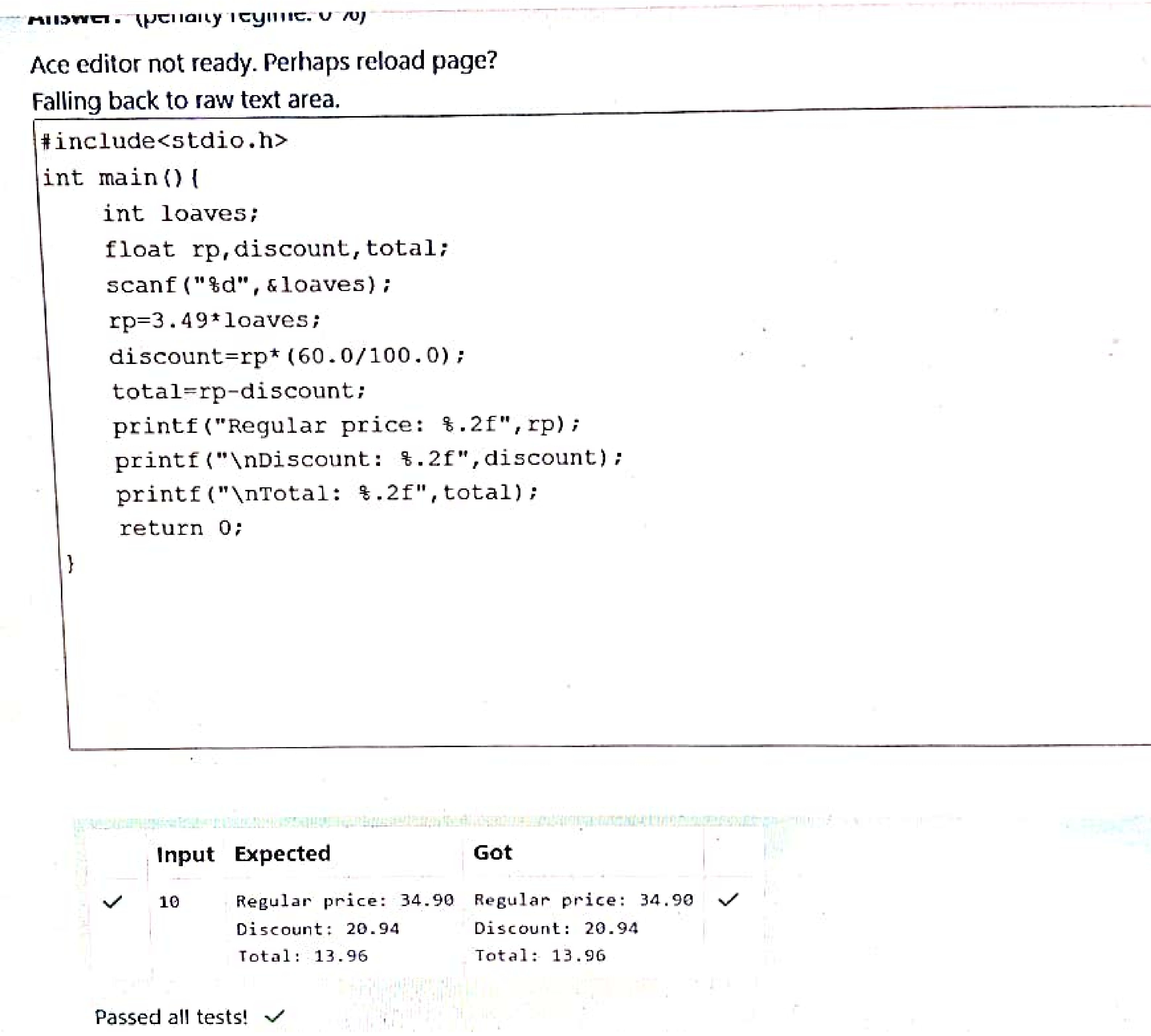


Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
#include<stdio.h>
int main(){
    int a;
    int b;
    scanf("%d %d",&a,&b);
    printf("%d",a+b);
    printf("\n%d",a-b);
    printf("\n%d",a*b);
    printf("\n%d",a/b);
    printf("\n%d",a%b);
    return 0;
}
```

	Input	Expected	Got	
✓	100 6	106 94 600 16 4	106 94 600 16 4	✓



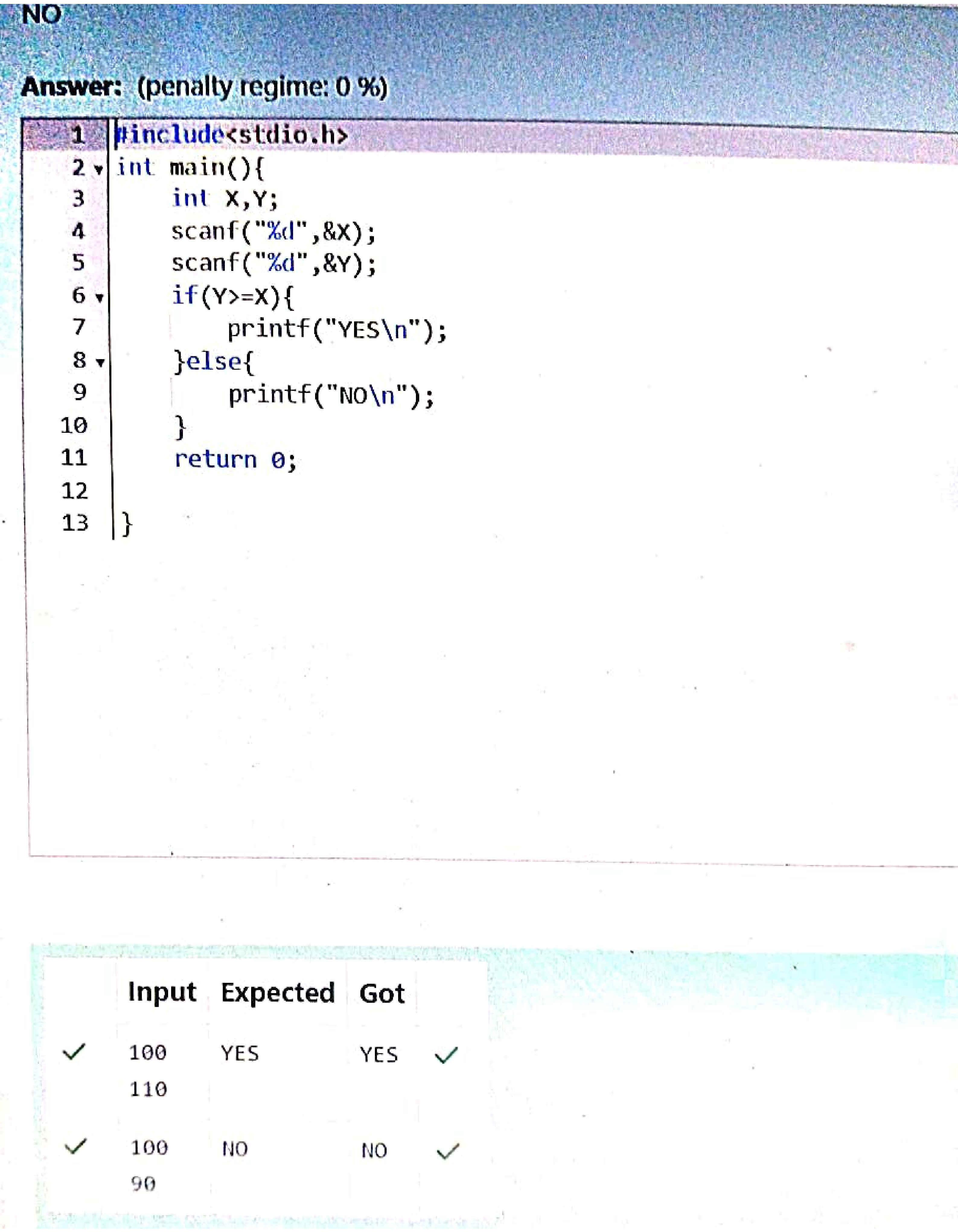
Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
#include<stdio.h>
int main(){
    int loaves;
    float rp,discount,total;
    scanf("%d",&loaves);
    rp=3.49*loaves;
    discount=rp*(60.0/100.0);
    total=rp-discount;
    printf("Regular price: %.2f",rp);
    printf("\nDiscount: %.2f",discount);
    printf("\nTotal: %.2f",total);
    return 0;
}
```

	Input	Expected	Got	
✓	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Regular price: 34.90 Discount: 20.94 Total: 13.96	✓

Passed all tests! ✓

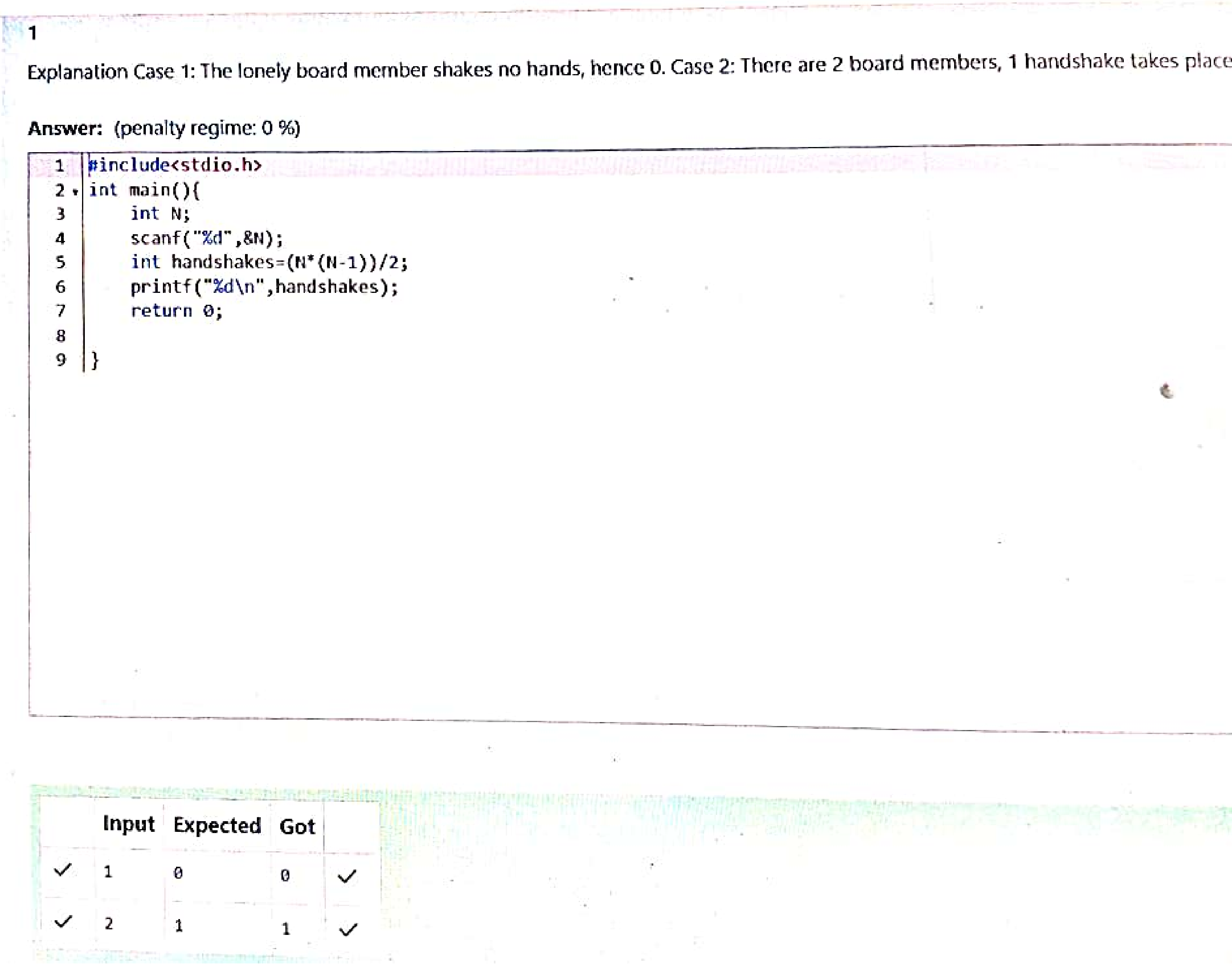


NO

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int X,Y;
    scanf("%d",&X);
    scanf("%d",&Y);
    if(Y>=X){
        printf("YES\n");
    }else{
        printf("NO\n");
    }
    return 0;

}
```

	Input	Expected	Got	
✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓



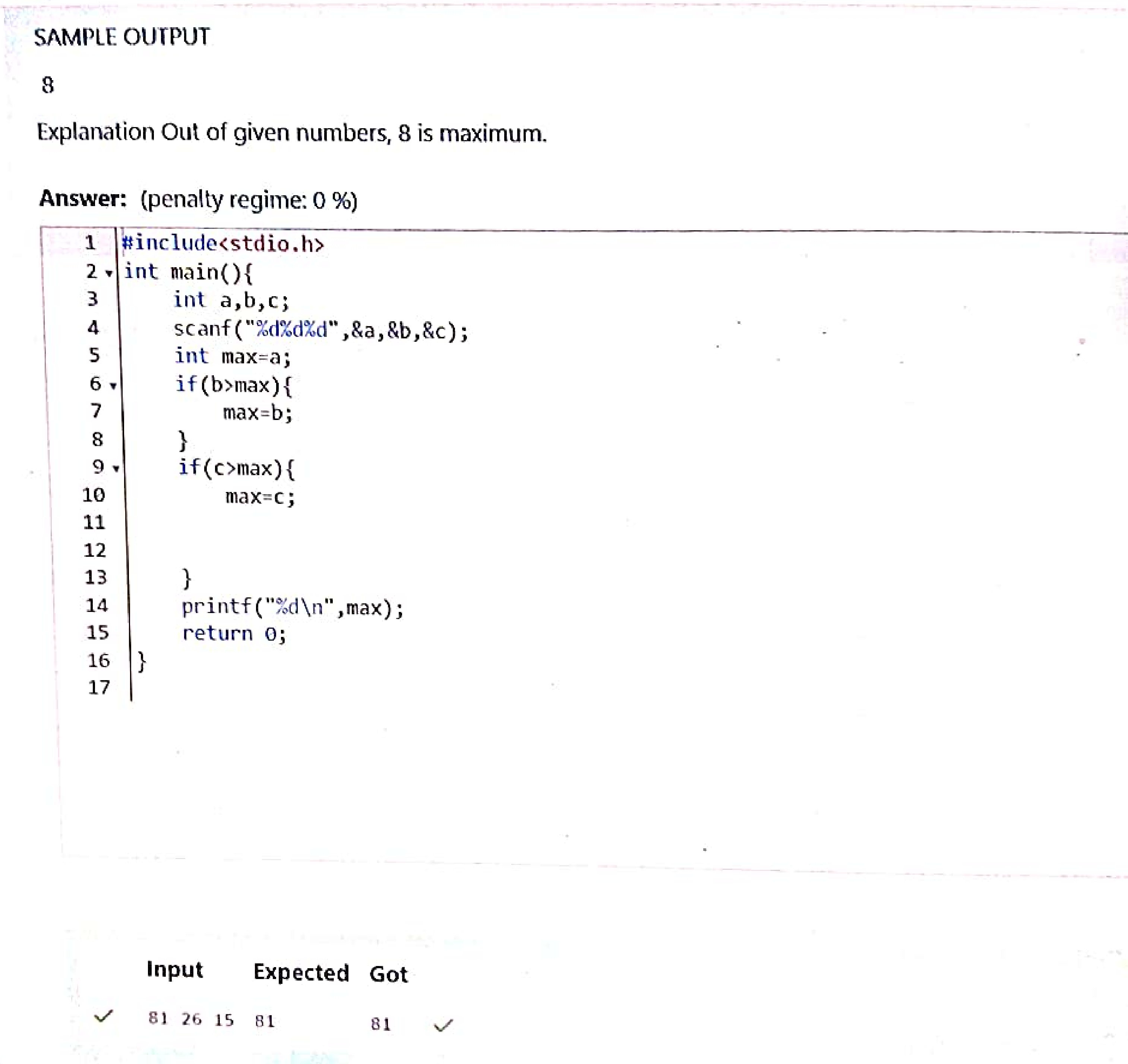
1

Explanation Case 1: The lonely board member shakes no hands, hence 0. Case 2: There are 2 board members, 1 handshake takes place

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int N;
    scanf("%d",&N);
    int handshakes=(N*(N-1))/2;
    printf("%d\n",handshakes);
    return 0;

}
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓



SAMPLE OUTPUT

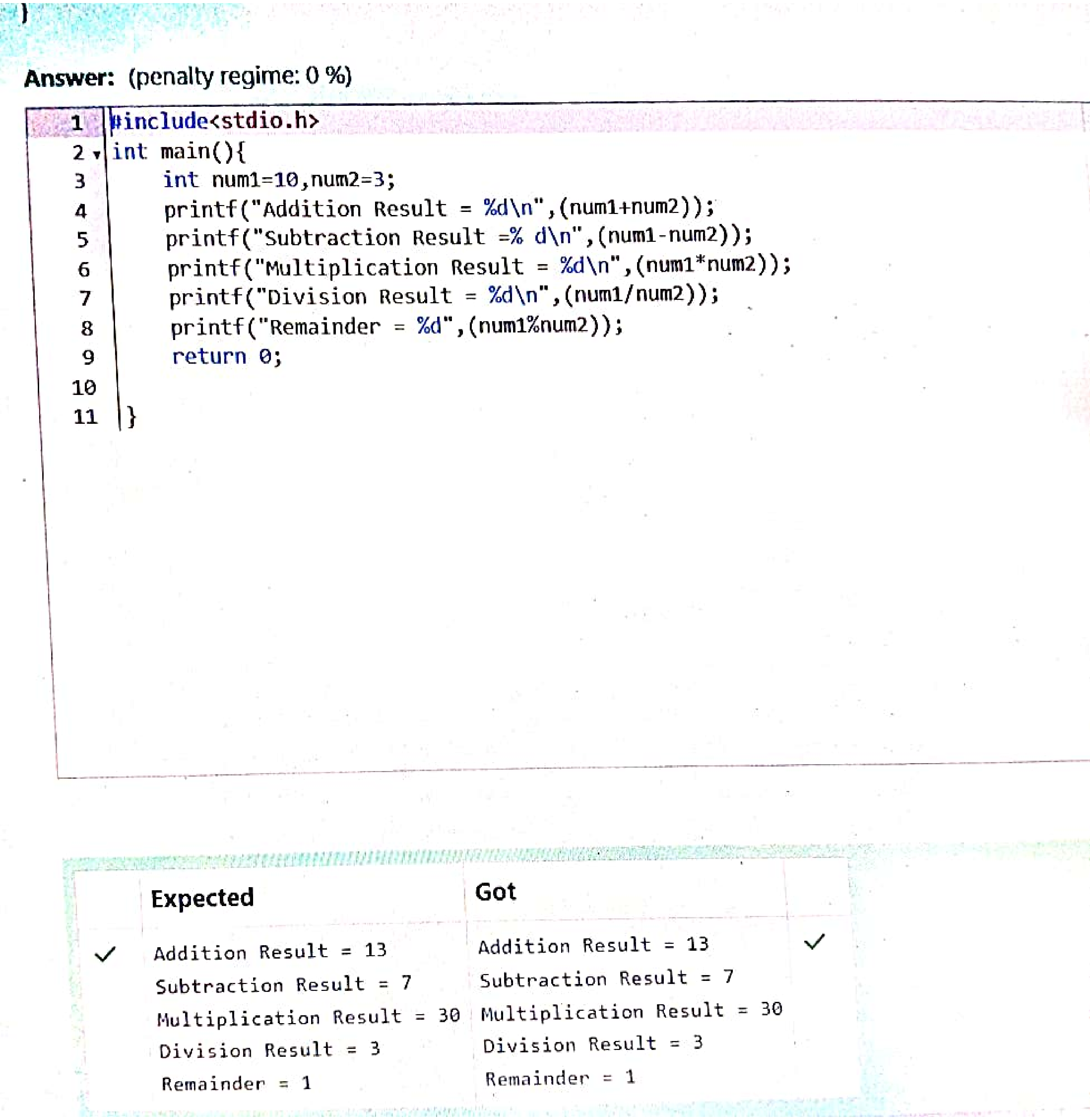
8

Explanation Out of given numbers, 8 is maximum.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    int max=a;
    if(b>max){
        max=b;
    }
    if(c>max){
        max=c;


    }
    printf("%d\n",max);
    return 0;
}

```

	Input	Expected	Got	
✓	81 26 15	81	81	✓

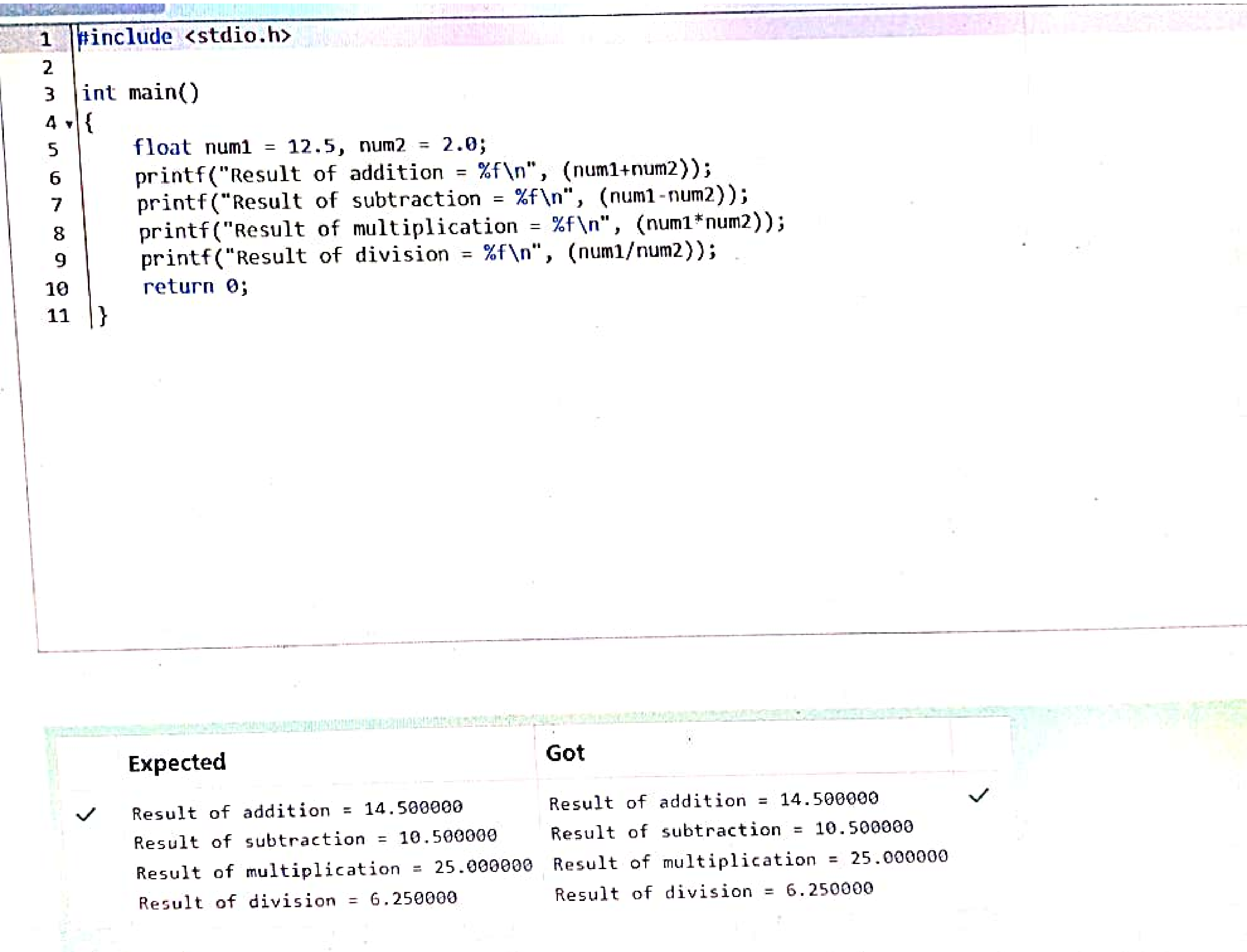


}

Answer: (penalty regime: 0 %)

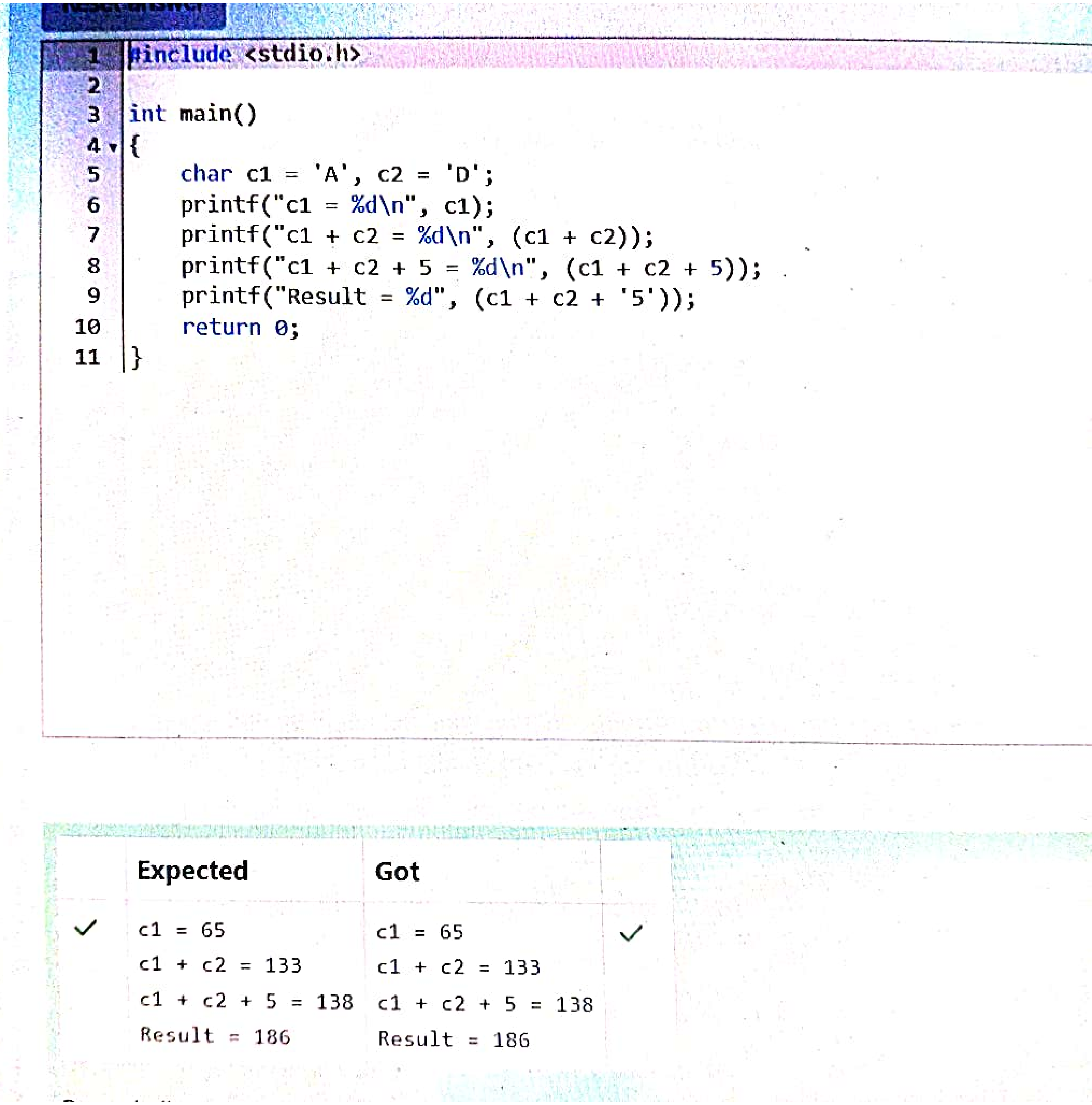
```c
#include<stdio.h>
int main(){
     int num1=10,num2=3;
     printf("Addition Result = %d\n",(num1+num2));
     printf("Subtraction Result =% d\n",(num1-num2));
     printf("Multiplication Result = %d\n",(num1*num2));
     printf("Division Result = %d\n",(num1/num2));
     printf("Remainder = %d",(num1%num2));
     return 0;

}
```

	Expected	Got	
✓	Addition Result = 13 Subtraction Result = 7 Multiplication Result = 30 Division Result = 3 Remainder = 1	Addition Result = 13 Subtraction Result = 7 Multiplication Result = 30 Division Result = 3 Remainder = 1	✓



```c
#include <stdio.h>

int main()
{
     float num1 = 12.5, num2 = 2.0;
     printf("Result of addition = %f\n", (num1+num2));
     printf("Result of subtraction = %f\n", (num1-num2));
     printf("Result of multiplication = %f\n", (num1*num2));
     printf("Result of division = %f\n", (num1/num2));
     return 0;
}
```

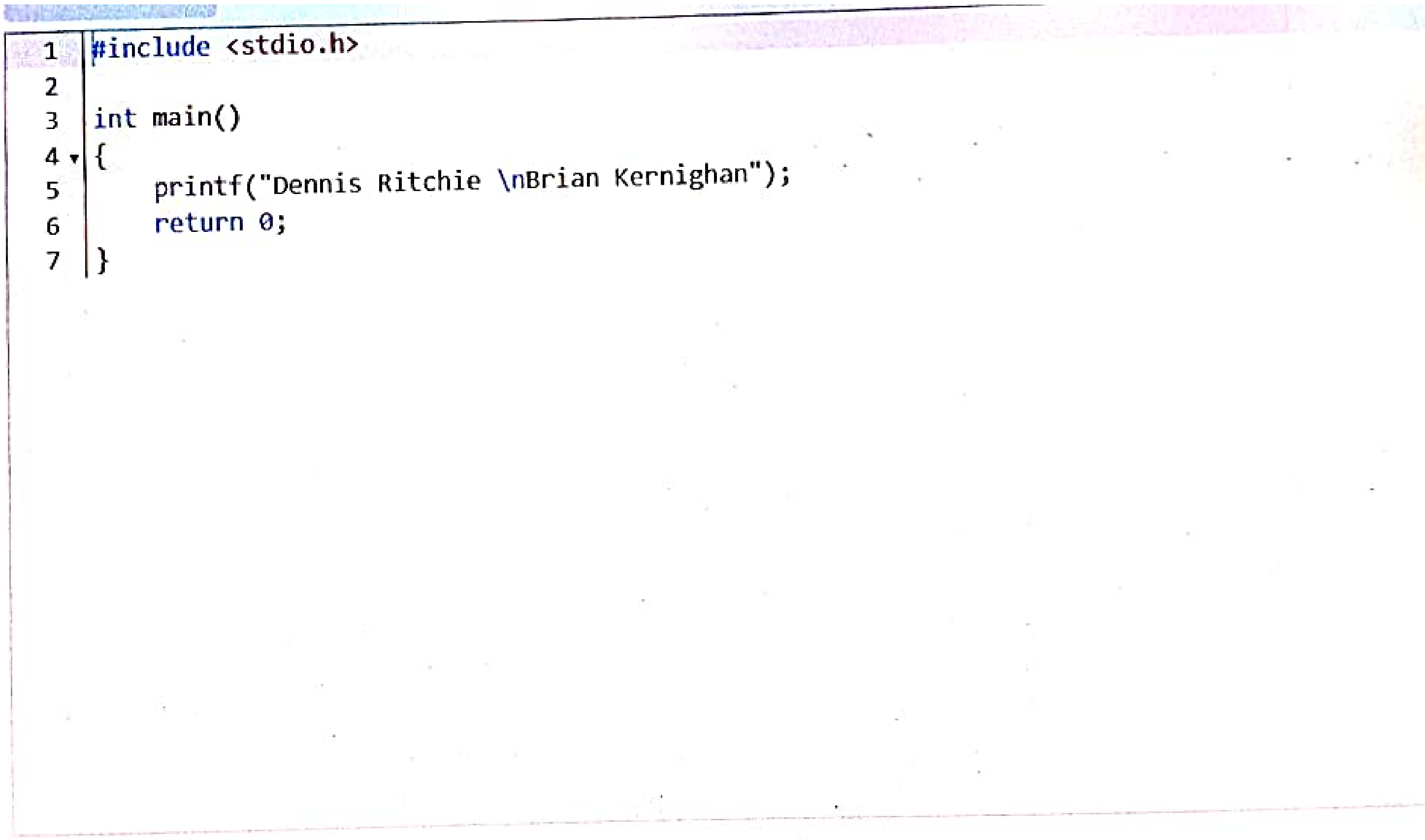
	Expected	Got	
✓	Result of addition = 14.500000 Result of subtraction = 10.500000 Result of multiplication = 25.000000 Result of division = 6.250000	Result of addition = 14.500000 Result of subtraction = 10.500000 Result of multiplication = 25.000000 Result of division = 6.250000	✓



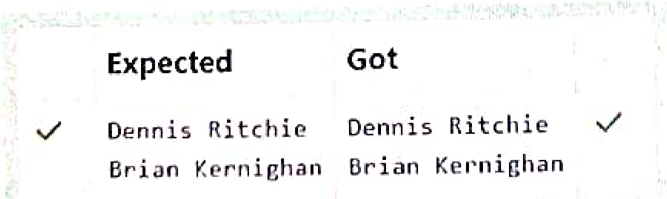
```c
#include <stdio.h>

int main()
{
    char c1 = 'A', c2 = 'D';
    printf("c1 = %d\n", c1);
    printf("c1 + c2 = %d\n", (c1 + c2));
    printf("c1 + c2 + 5 = %d\n", (c1 + c2 + 5));
    printf("Result = %d", (c1 + c2 + '5'));
    return 0;
}
```

	Expected	Got	

✓	c1 = 65
	c1 + c2 = 133
	c1 + c2 + 5 = 138
	Result = 186
		c1 = 65
		c1 + c2 = 133
		c1 + c2 + 5 = 138
		Result = 186
			✓



```
#include <stdio.h>

int main()
{
    printf("Dennis Ritchie \nBrian Kernighan");
    return 0;
}
```



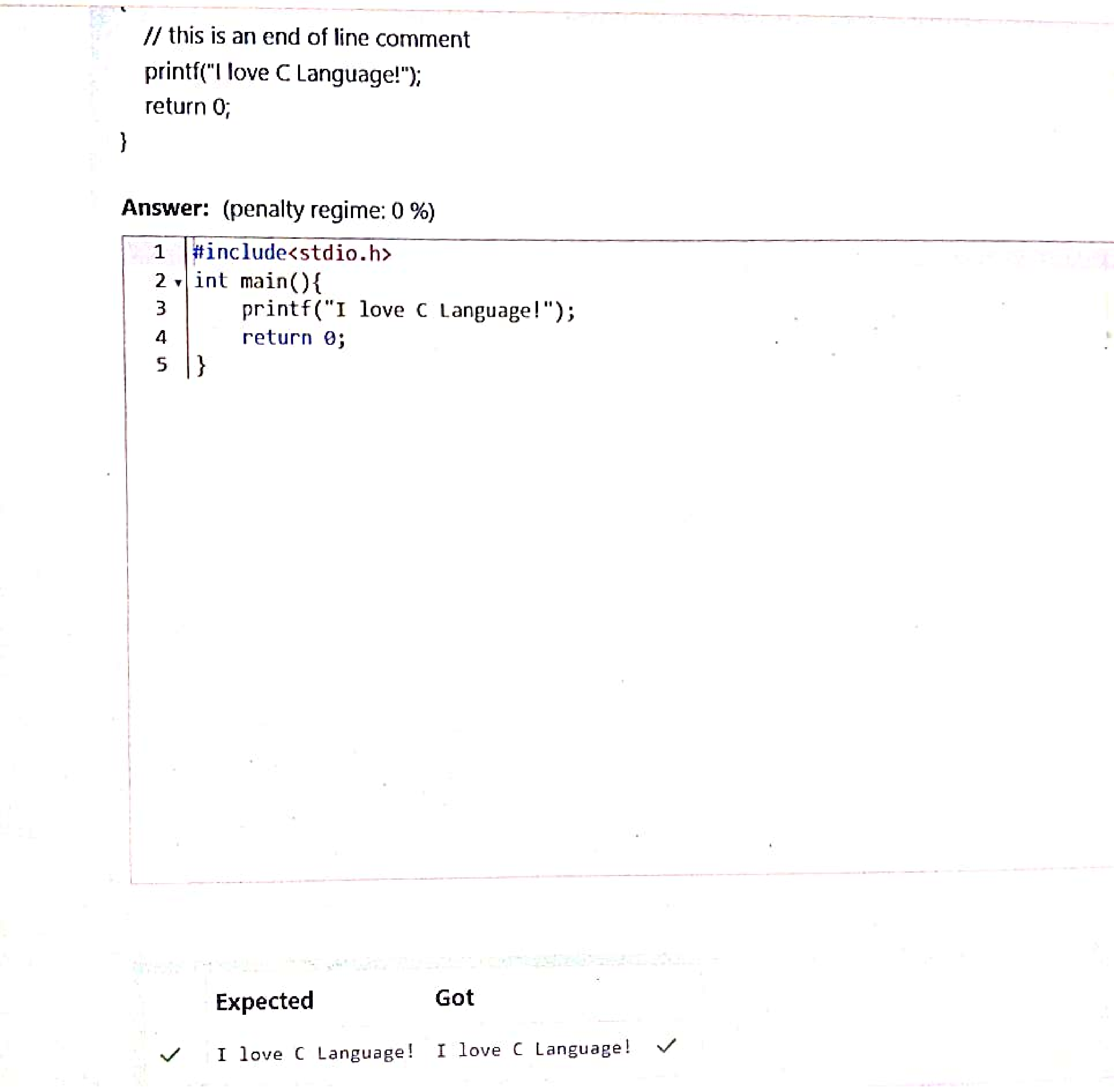
	Expected	Got	
✓	Dennis Ritchie Brian Kernighan	Dennis Ritchie Brian Kernighan	✓

```
    printf("Four\nFive\n");
    return 0;
}
```

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    printf("One Two");
    printf("Three\n");
    printf("Four\nFive\n");
    return 0;
}
```

	Expected	Got	
✓	One TwoThree Four Five	One TwoThree Four Five	✓

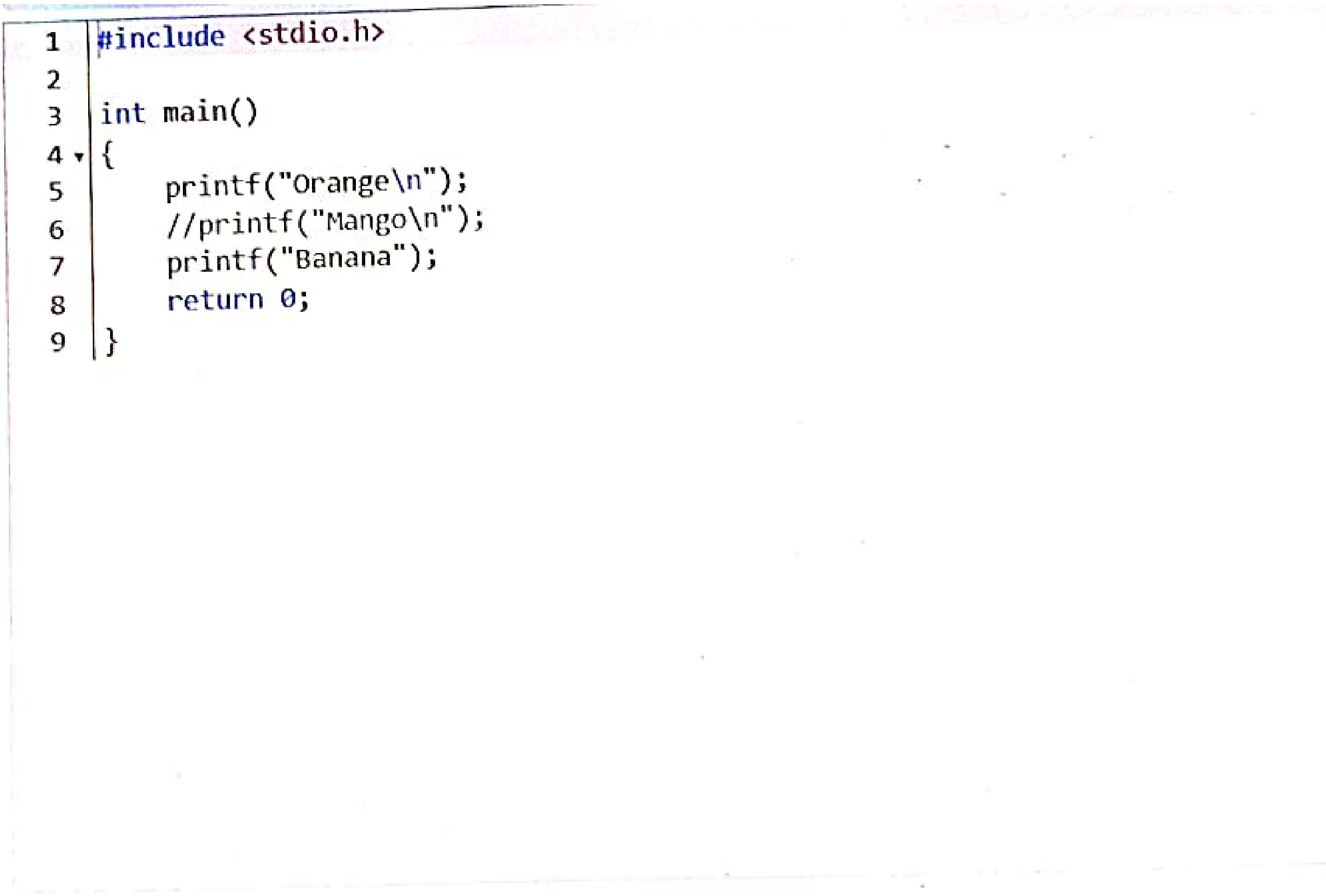


```
    // this is an end of line comment
    printf("I love C Language!");
    return 0;
}
```

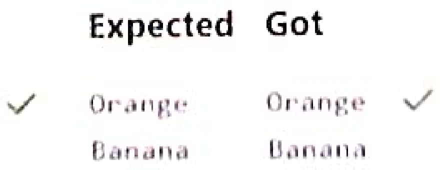
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    printf("I love C Language!");
    return 0;
}
```

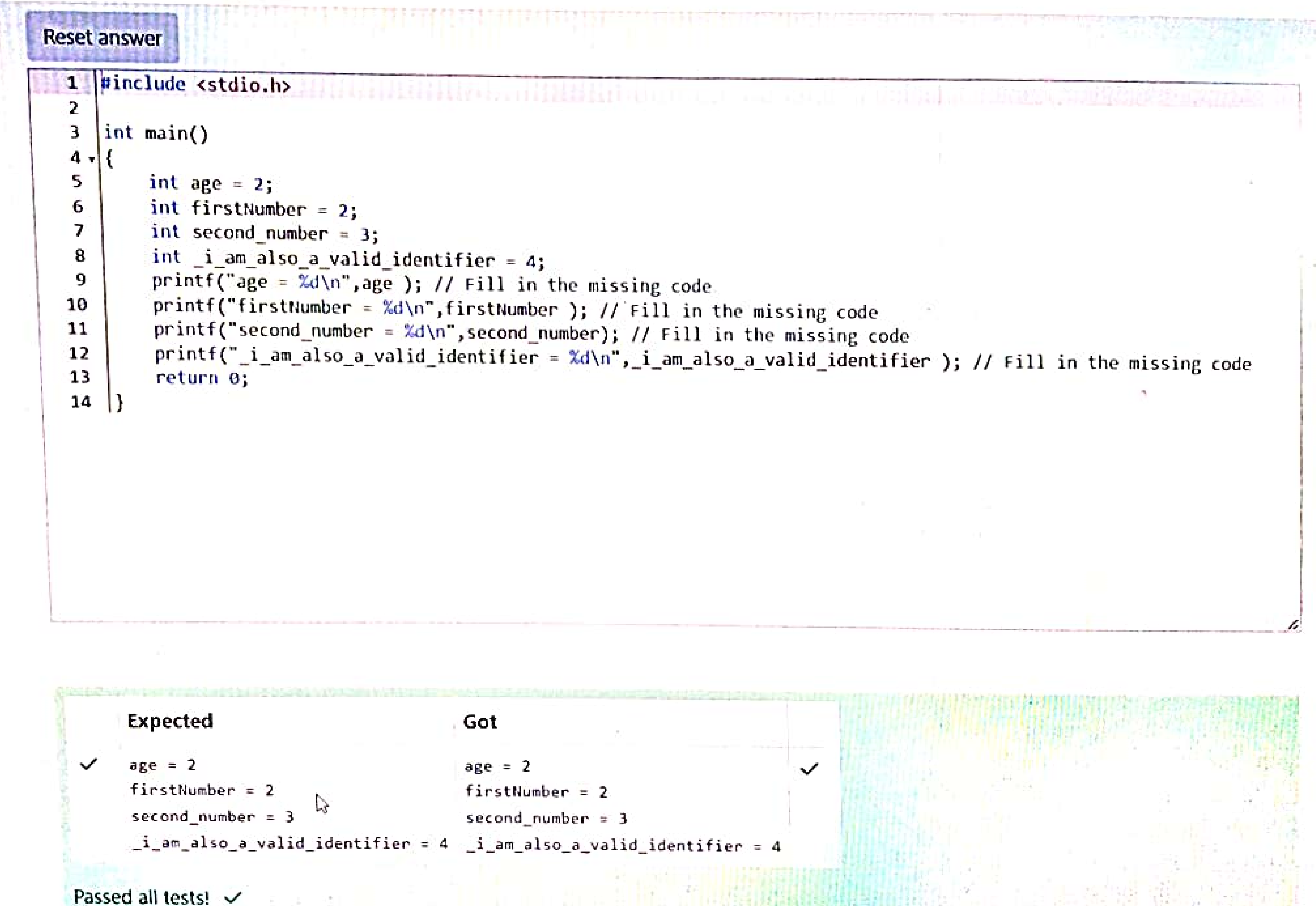
	Expected	Got	
✓	`I love C Language!`	`I love C Language!`	✓



```c
#include <stdio.h>

int main()
{
    printf("Orange\n");
    //printf("Mango\n");
    printf("Banana");
    return 0;
}
```



	Expected	Got	
✓	Orange Banana	Orange Banana	✓

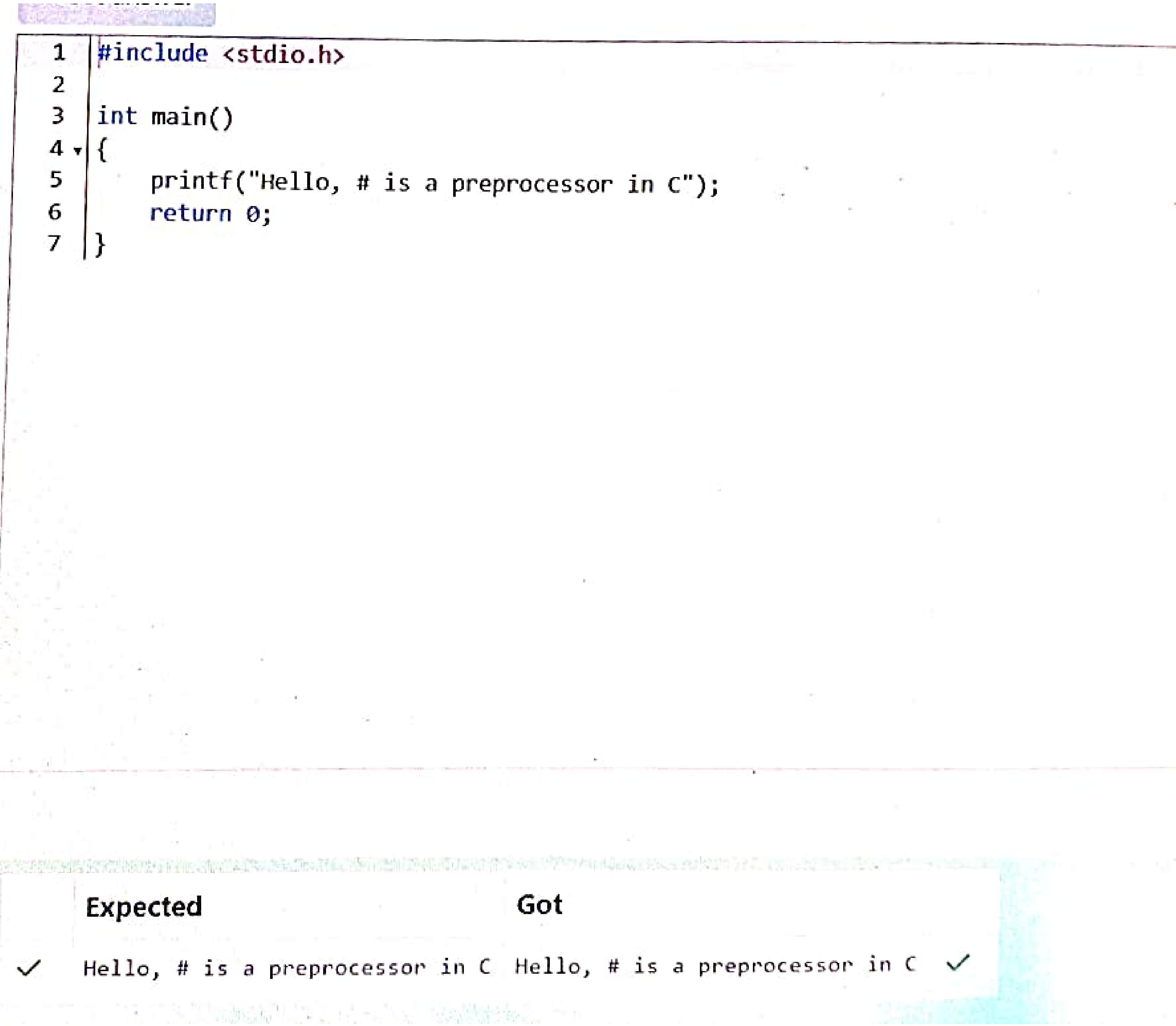


```c
#include <stdio.h>

int main()
{
    int age = 2;
    int firstNumber = 2;
    int second_number = 3;
    int _i_am_also_a_valid_identifier = 4;
    printf("age = %d\n",age ); // Fill in the missing code
    printf("firstNumber = %d\n",firstNumber ); // Fill in the missing code
    printf("second_number = %d\n",second_number); // Fill in the missing code
    printf("_i_am_also_a_valid_identifier = %d\n",_i_am_also_a_valid_identifier ); // Fill in the missing code
    return 0;
}
```

	Expected	Got	

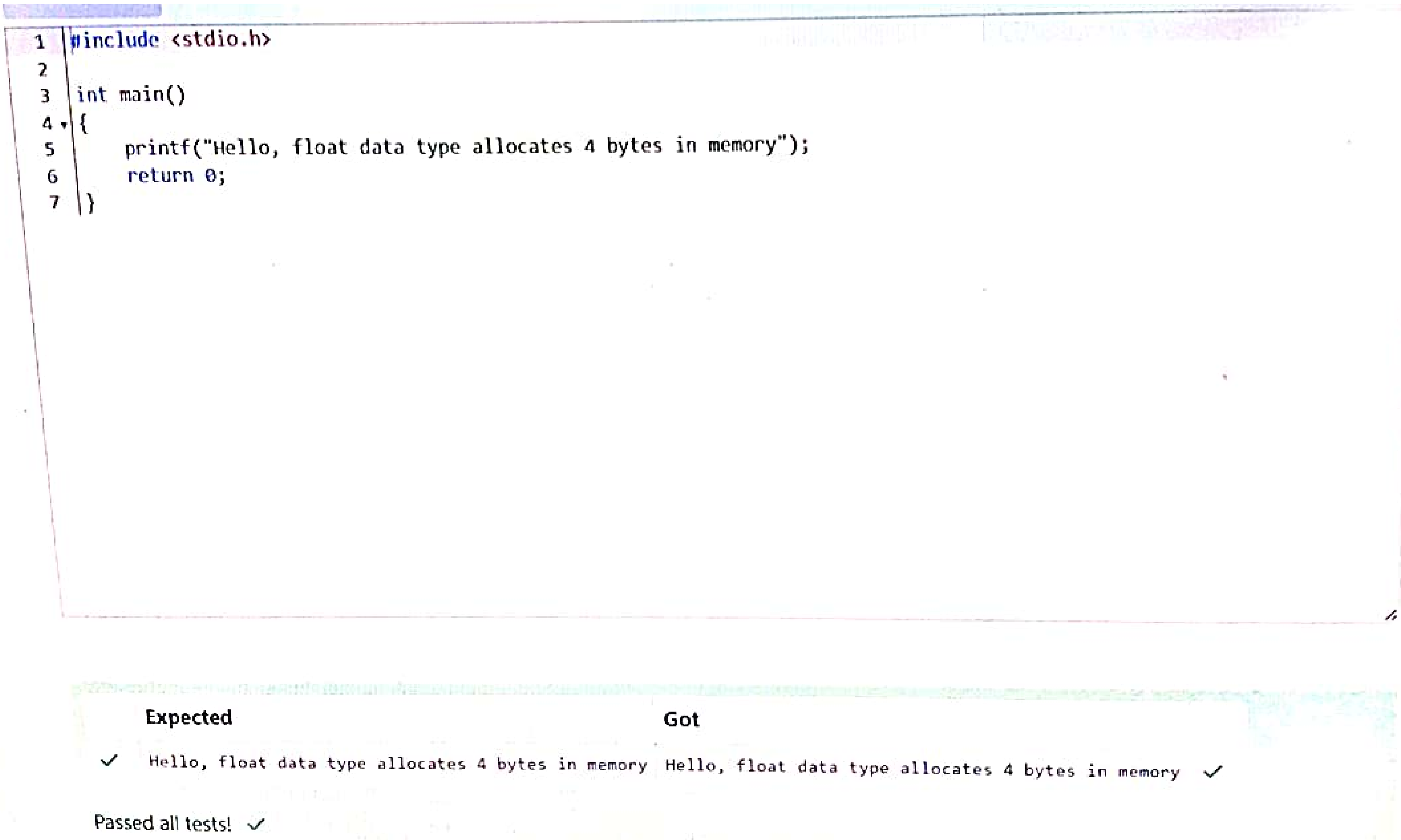
✓	age = 2
	firstNumber = 2
	second_number = 3
	_i_am_also_a_valid_identifier = 4
		age = 2
		firstNumber = 2
		second_number = 3
		_i_am_also_a_valid_identifier = 4
			✓

Passed all tests! ✓



```
#include <stdio.h>

int main()
{
    printf("Hello, # is a preprocessor in C");
    return 0;
}
```

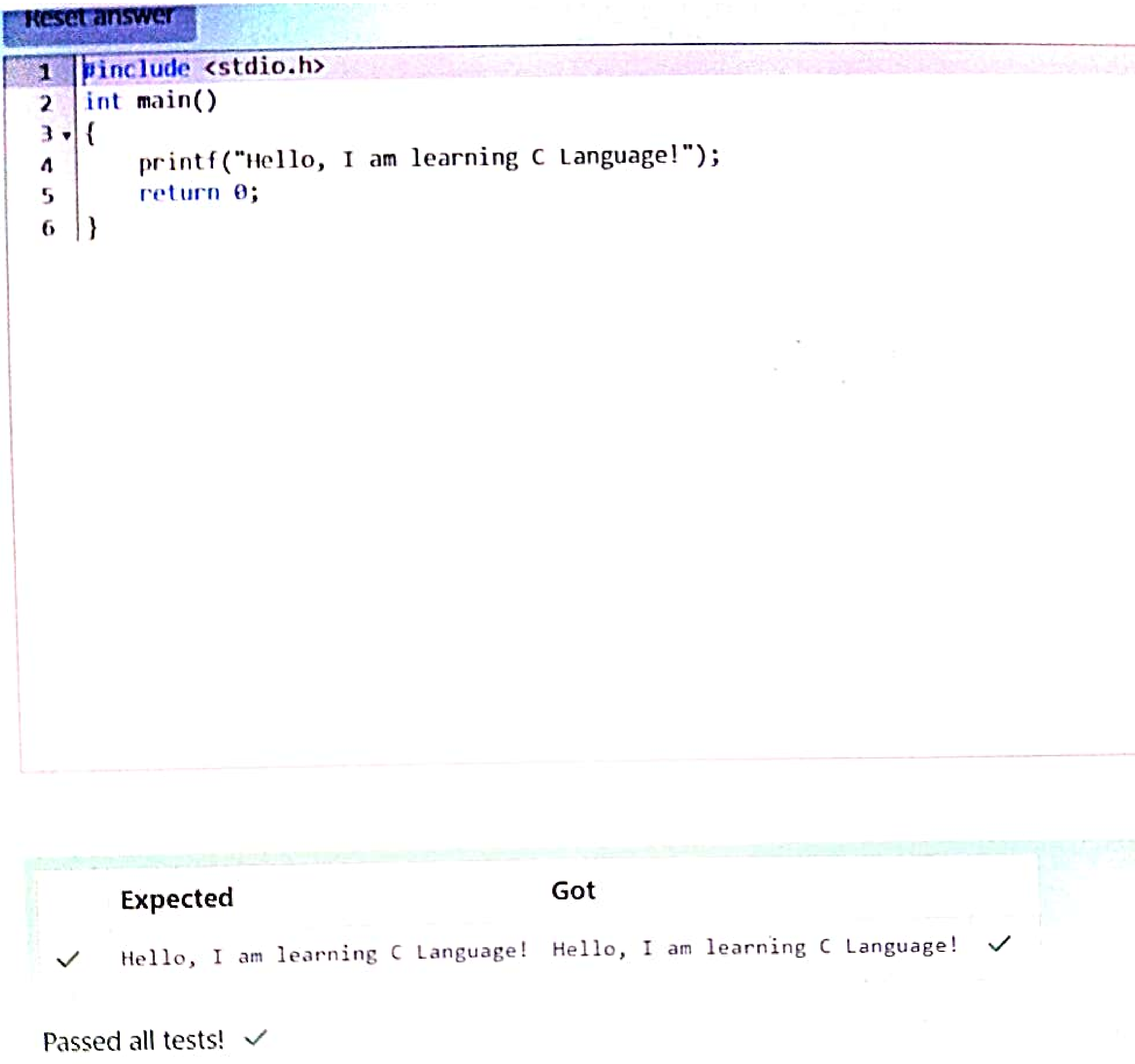
	Expected	Got	
✓	Hello, # is a preprocessor in C	Hello, # is a preprocessor in C	✓



```c
#include <stdio.h>

int main()
{
    printf("Hello, float data type allocates 4 bytes in memory");
    return 0;
}
```

	Expected	Got	
✓	Hello, float data type allocates 4 bytes in memory	Hello, float data type allocates 4 bytes in memory	✓

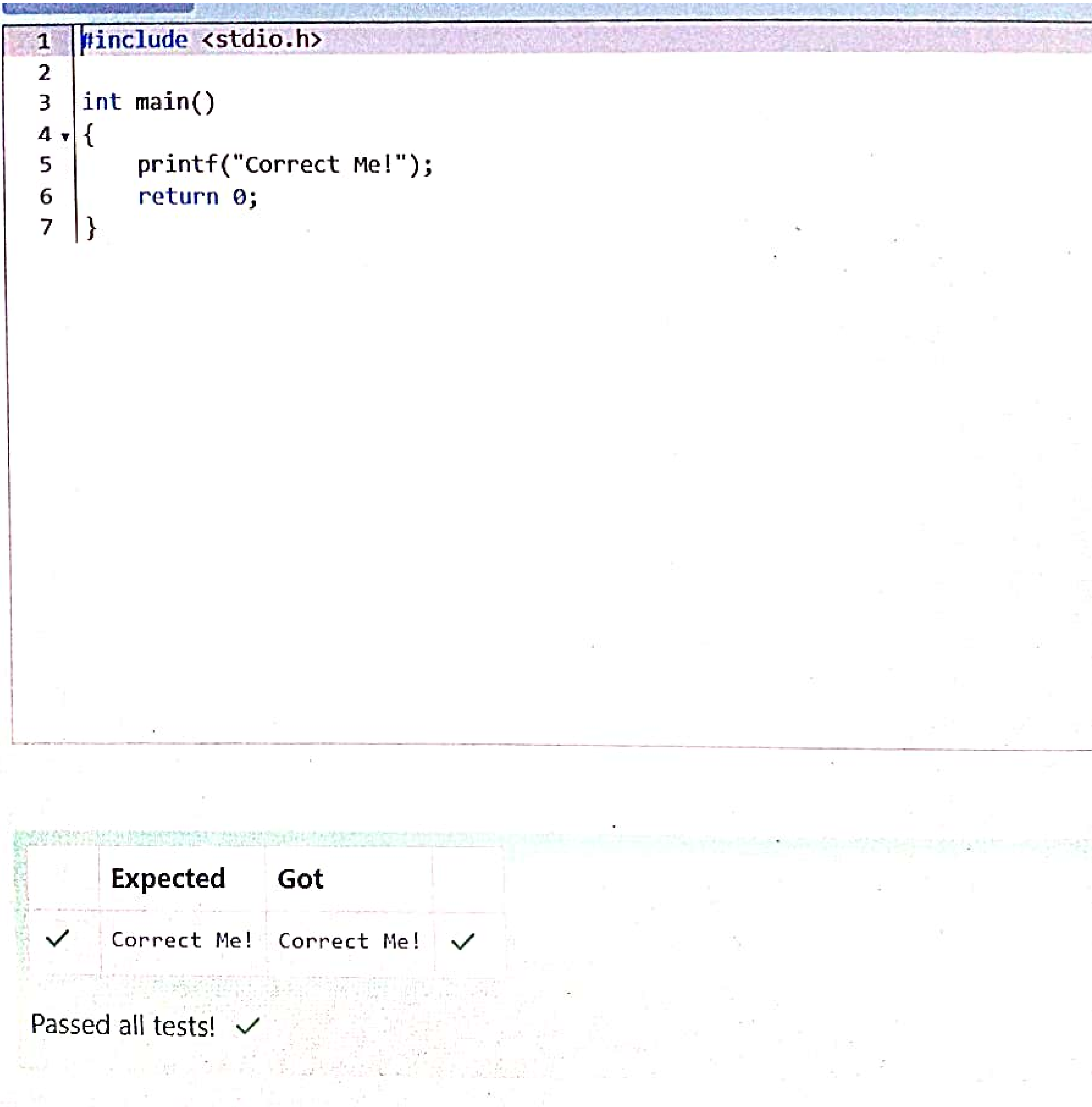
Passed all tests! ✓



```c
#include <stdio.h>
int main()
{
    printf("Hello, I am learning C Language!");
    return 0;
}
```

	Expected	Got	
✓	Hello, I am learning C Language!	Hello, I am learning C Language!	✓

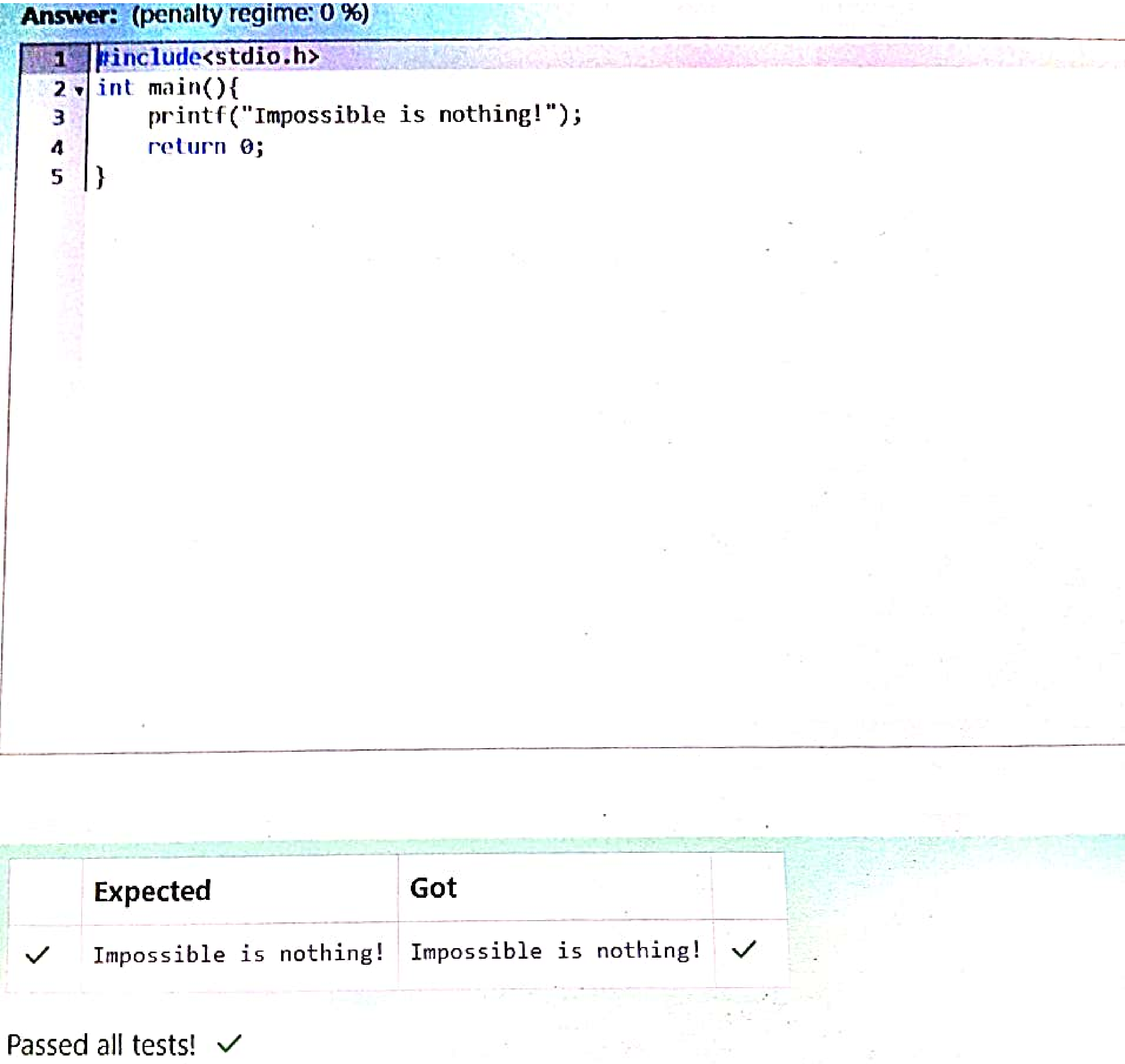
Passed all tests! ✓



```c
#include <stdio.h>

int main()
{
    printf("Correct Me!");
    return 0;
}
```

	Expected	Got	
✓	Correct Me!	Correct Me!	✓

Passed all tests! ✓

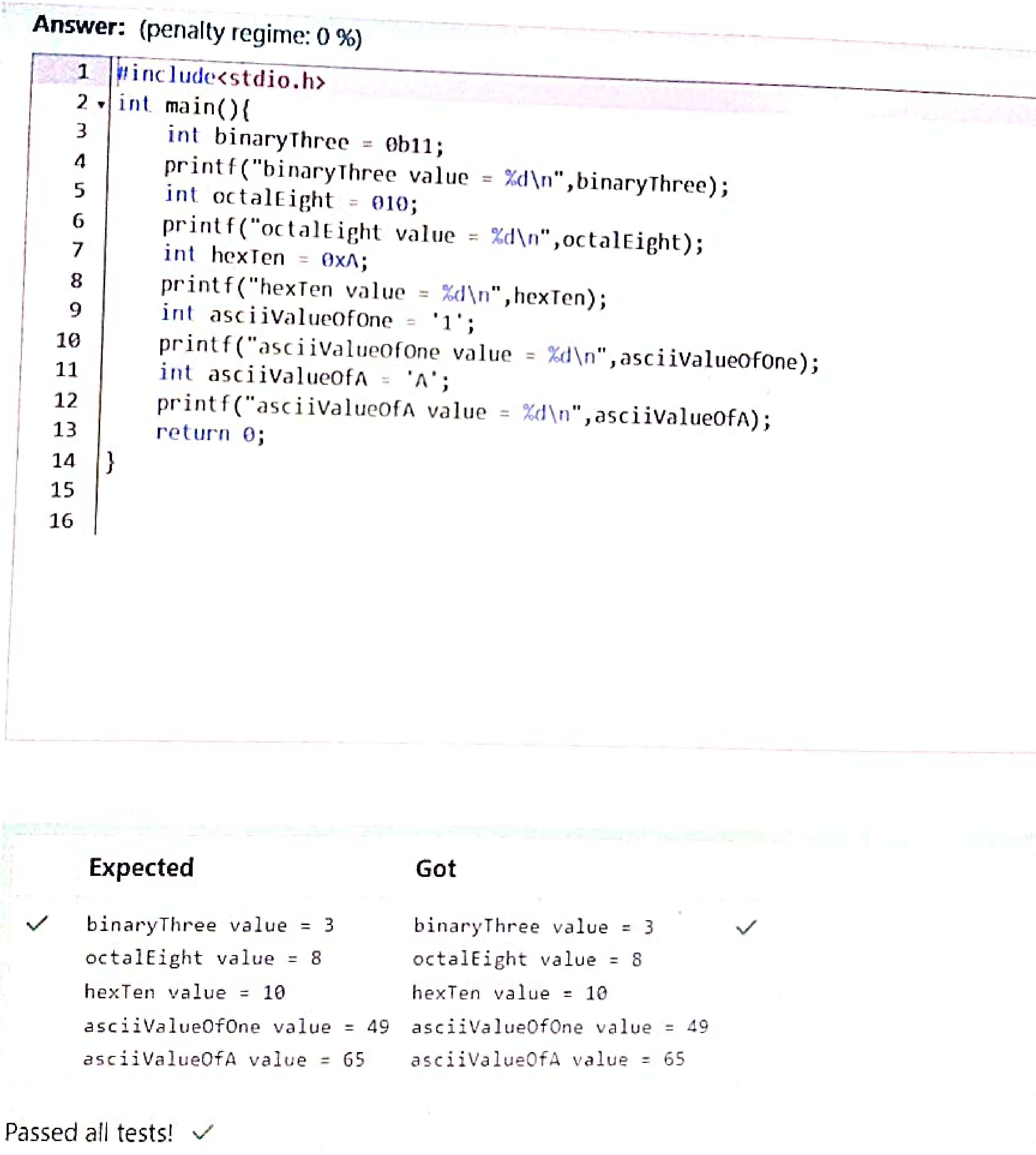


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    printf("Impossible is nothing!");
    return 0;
}
```

	Expected	Got	
✓	`Impossible is nothing!`	`Impossible is nothing!`	✓

Passed all tests! ✓

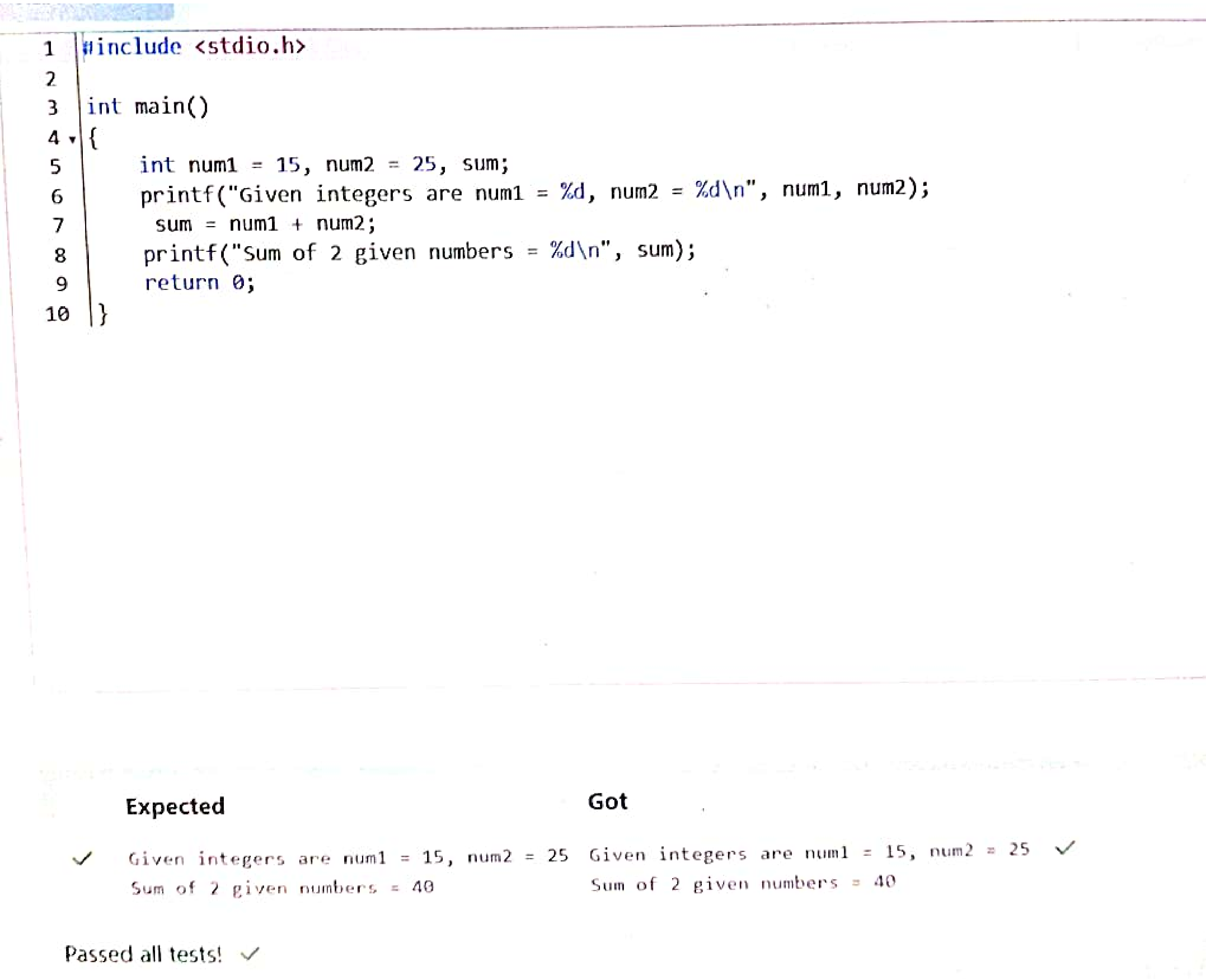


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int binaryThree = 0b11;
    printf("binaryThree value = %d\n",binaryThree);
    int octalEight = 010;
    printf("octalEight value = %d\n",octalEight);
    int hexTen = 0xA;
    printf("hexTen value = %d\n",hexTen);
    int asciiValueOfOne = '1';
    printf("asciiValueOfOne value = %d\n",asciiValueOfOne);
    int asciiValueOfA = 'A';
    printf("asciiValueOfA value = %d\n",asciiValueOfA);
    return 0;
}
```

	Expected	Got	
✓	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	✓

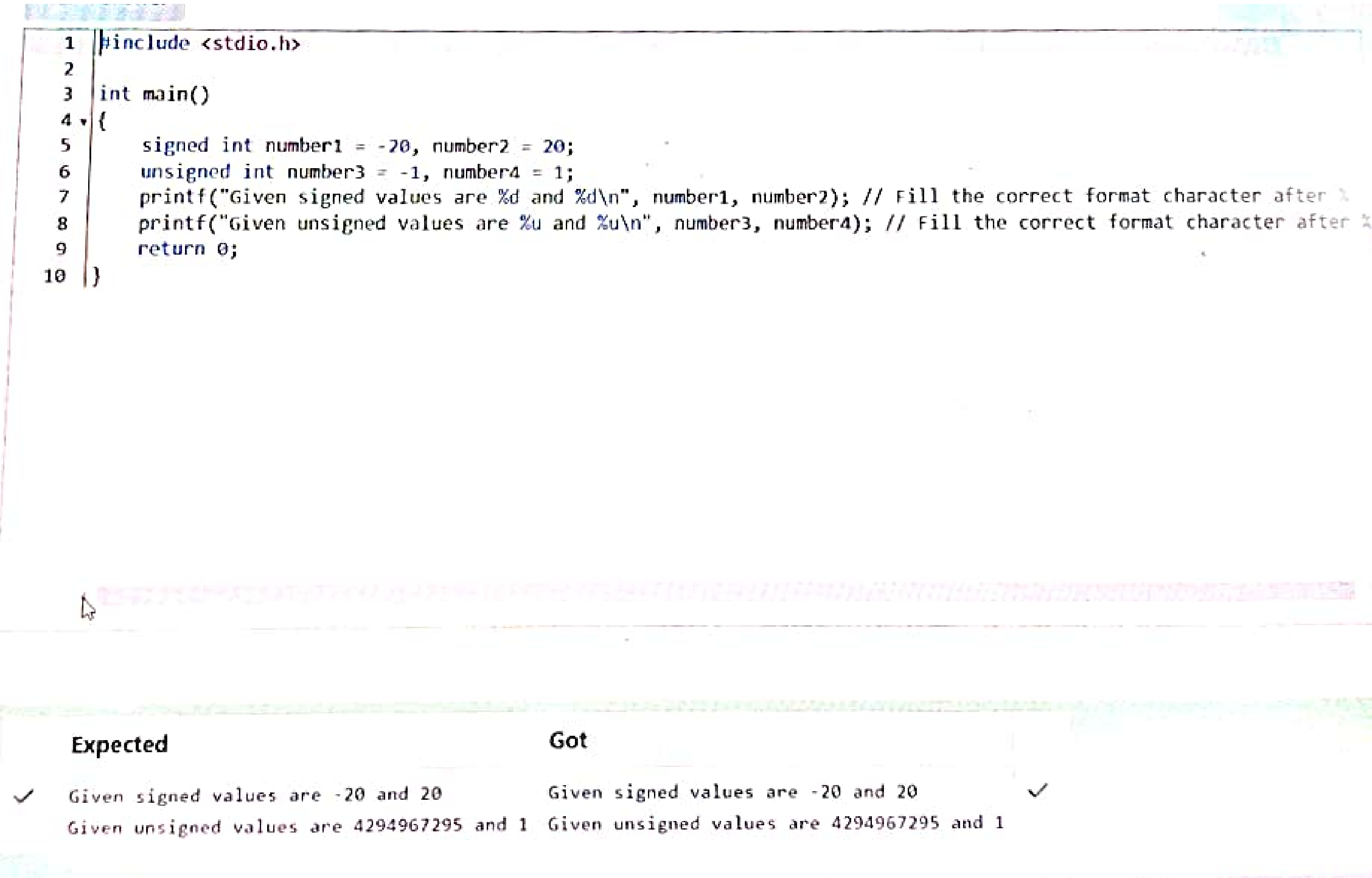
Passed all tests! ✓



```c
#include <stdio.h>

int main()
{
     int num1 = 15, num2 = 25, sum;
     printf("Given integers are num1 = %d, num2 = %d\n", num1, num2);
      sum = num1 + num2;
     printf("Sum of 2 given numbers = %d\n", sum);
     return 0;
}
```

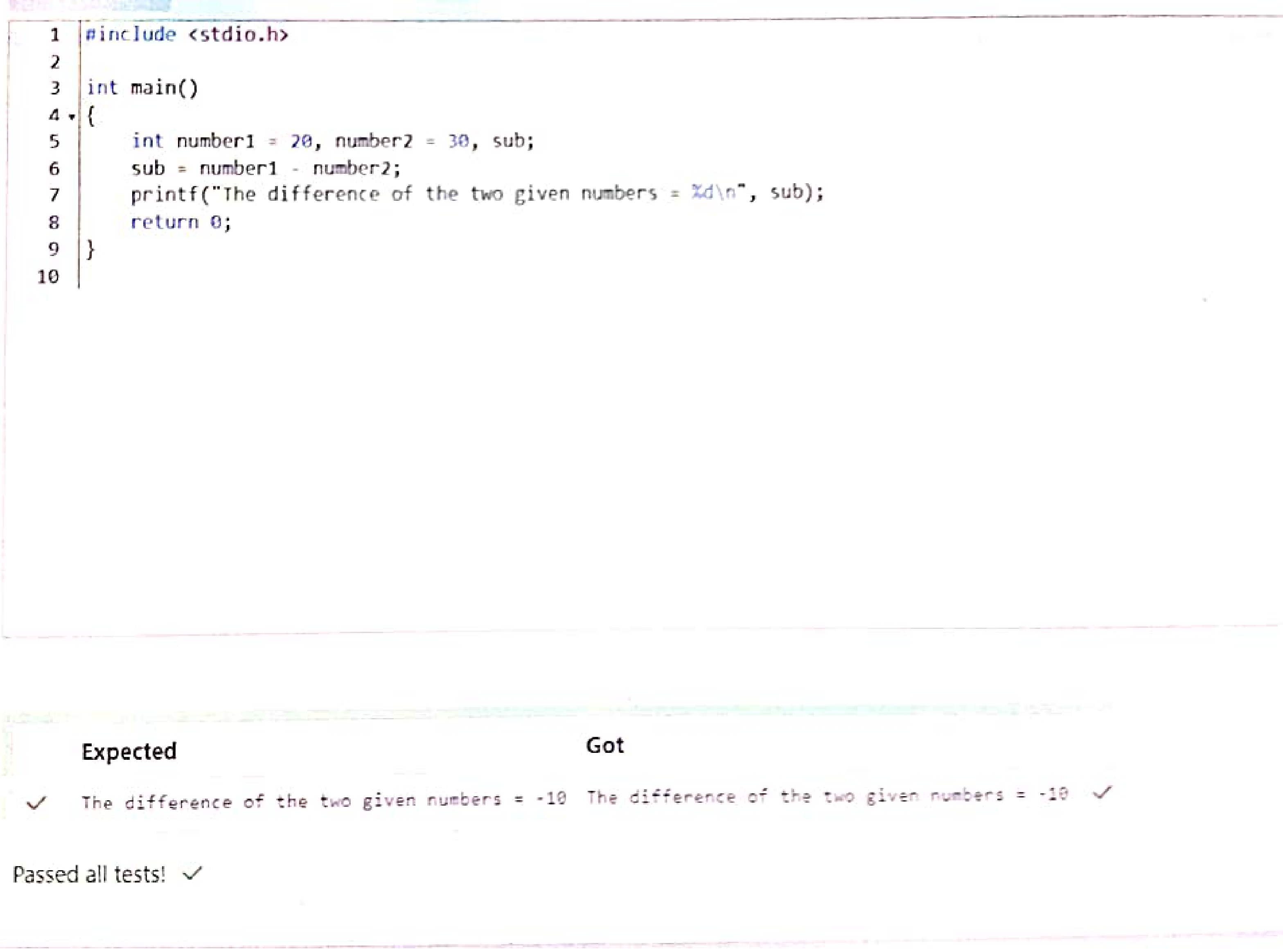
	Expected	Got	
✓	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	✓

Passed all tests! ✓



```c
#include <stdio.h>

int main()
{
    signed int number1 = -20, number2 = 20;
    unsigned int number3 = -1, number4 = 1;
    printf("Given signed values are %d and %d\n", number1, number2); // Fill the correct format character after %
    printf("Given unsigned values are %u and %u\n", number3, number4); // Fill the correct format character after %
    return 0;
}
```

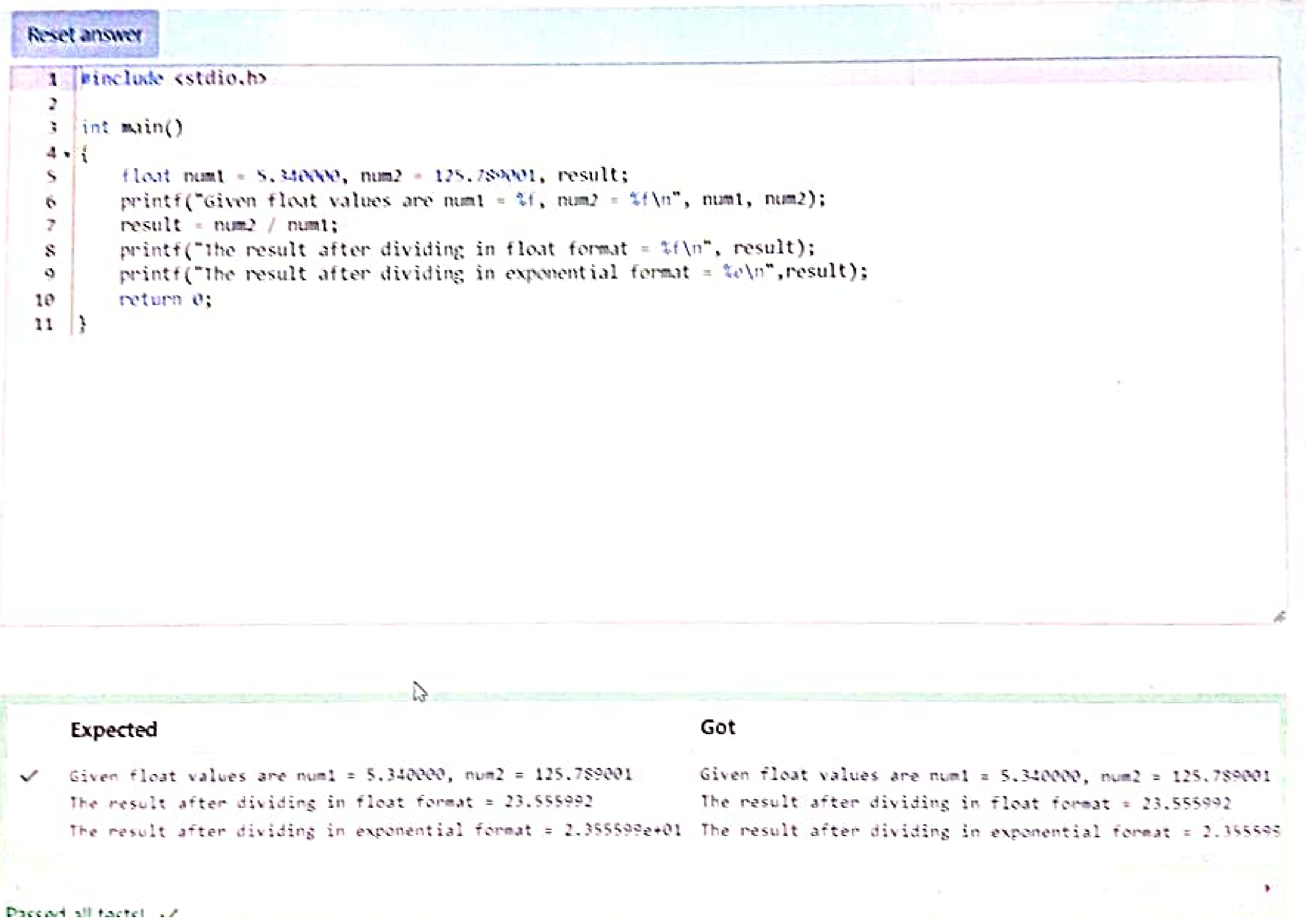
	Expected	Got	
✓	Given signed values are -20 and 20 Given unsigned values are 4294967295 and 1	Given signed values are -20 and 20 Given unsigned values are 4294967295 and 1	✓



```c
#include <stdio.h>

int main()
{
    int number1 = 20, number2 = 30, sub;
    sub = number1 - number2;
    printf("The difference of the two given numbers = %d\n", sub);
    return 0;
}

```

	Expected	Got	
✓	The difference of the two given numbers = -10	The difference of the two given numbers = -10	✓

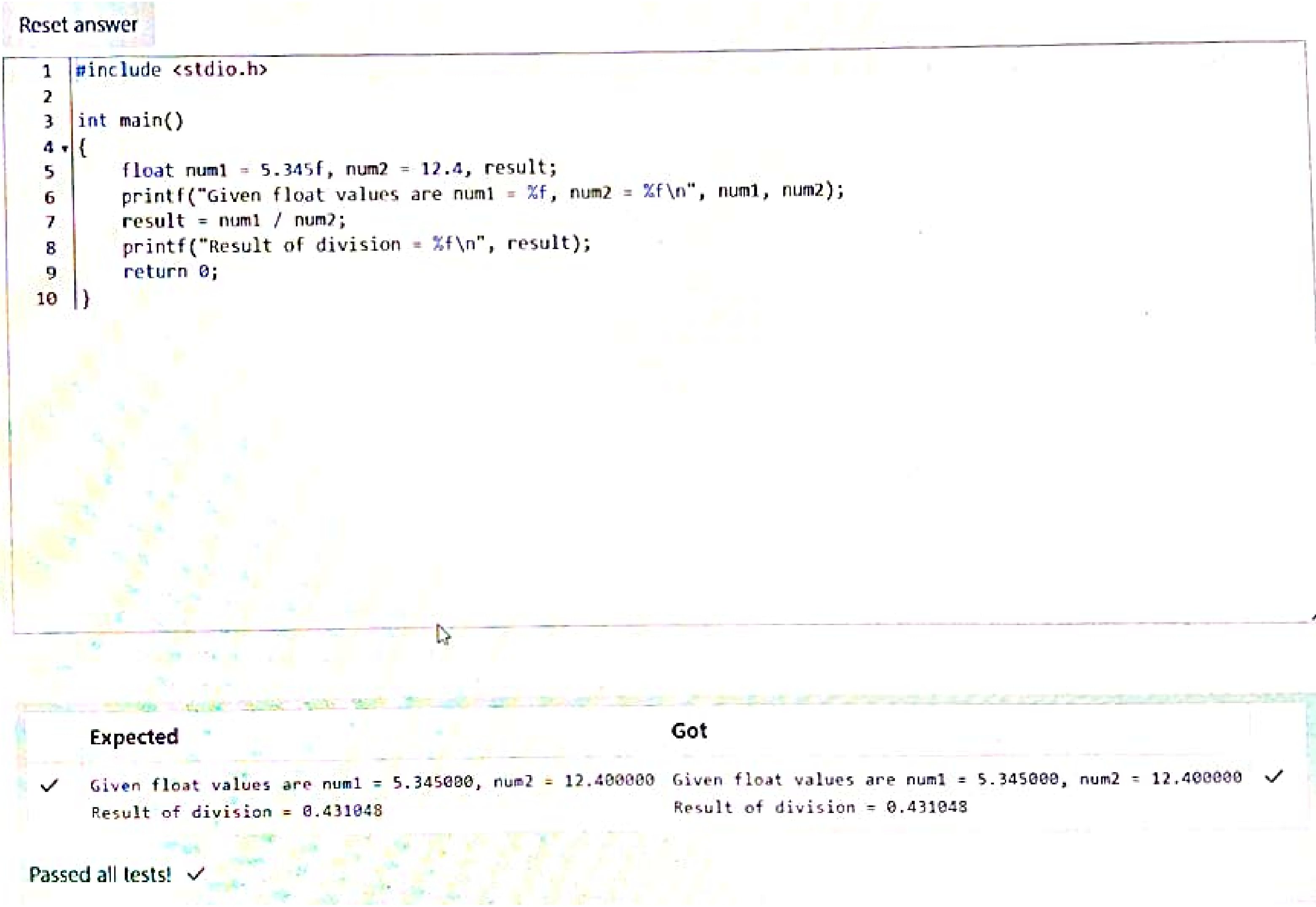
Passed all tests! ✓



```c
#include <stdio.h>

int main()
{
    float num1 = 5.340000, num2 = 125.789001, result;
    printf("Given float values are num1 = %f, num2 = %f\n", num1, num2);
    result = num2 / num1;
    printf("The result after dividing in float format = %f\n", result);
    printf("The result after dividing in exponential format = %e\n",result);
    return 0;
}
```

	Expected	Got
✓	Given float values are num1 = 5.340000, num2 = 125.789001 The result after dividing in float format = 23.555992 The result after dividing in exponential format = 2.355599e+01	Given float values are num1 = 5.340000, num2 = 125.789001 The result after dividing in float format = 23.555992 The result after dividing in exponential format = 2.355595

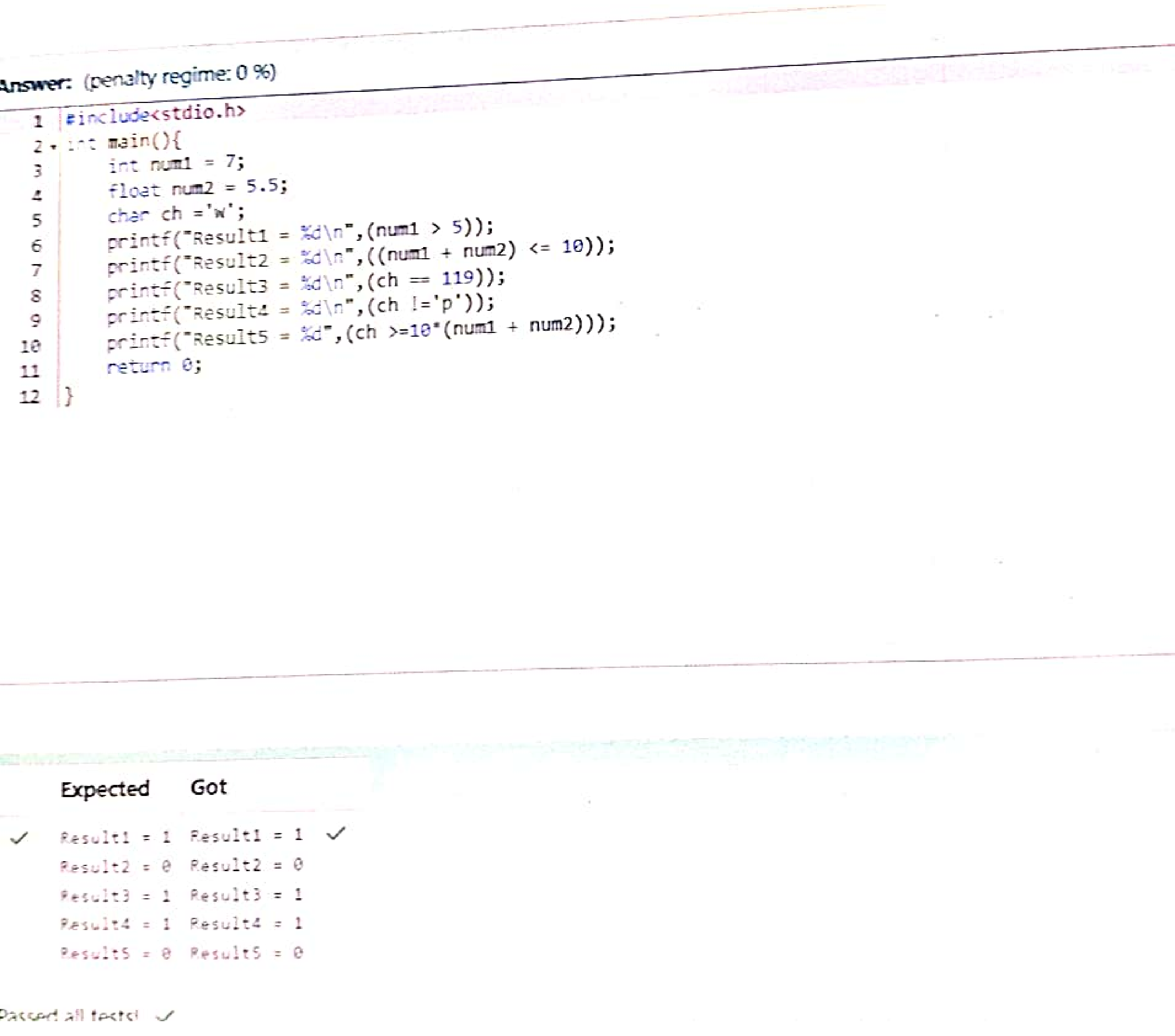
Passed all tests! ✓



```c
#include <stdio.h>

int main()
{
    float num1 = 5.345f, num2 = 12.4, result;
    printf("Given float values are num1 = %f, num2 = %f\n", num1, num2);
    result = num1 / num2;
    printf("Result of division = %f\n", result);
    return 0;
}
```

	Expected	Got	
✓	Given float values are num1 = 5.345000, num2 = 12.400000 Result of division = 0.431048	Given float values are num1 = 5.345000, num2 = 12.400000 Result of division = 0.431048	✓

Passed all tests! ✓



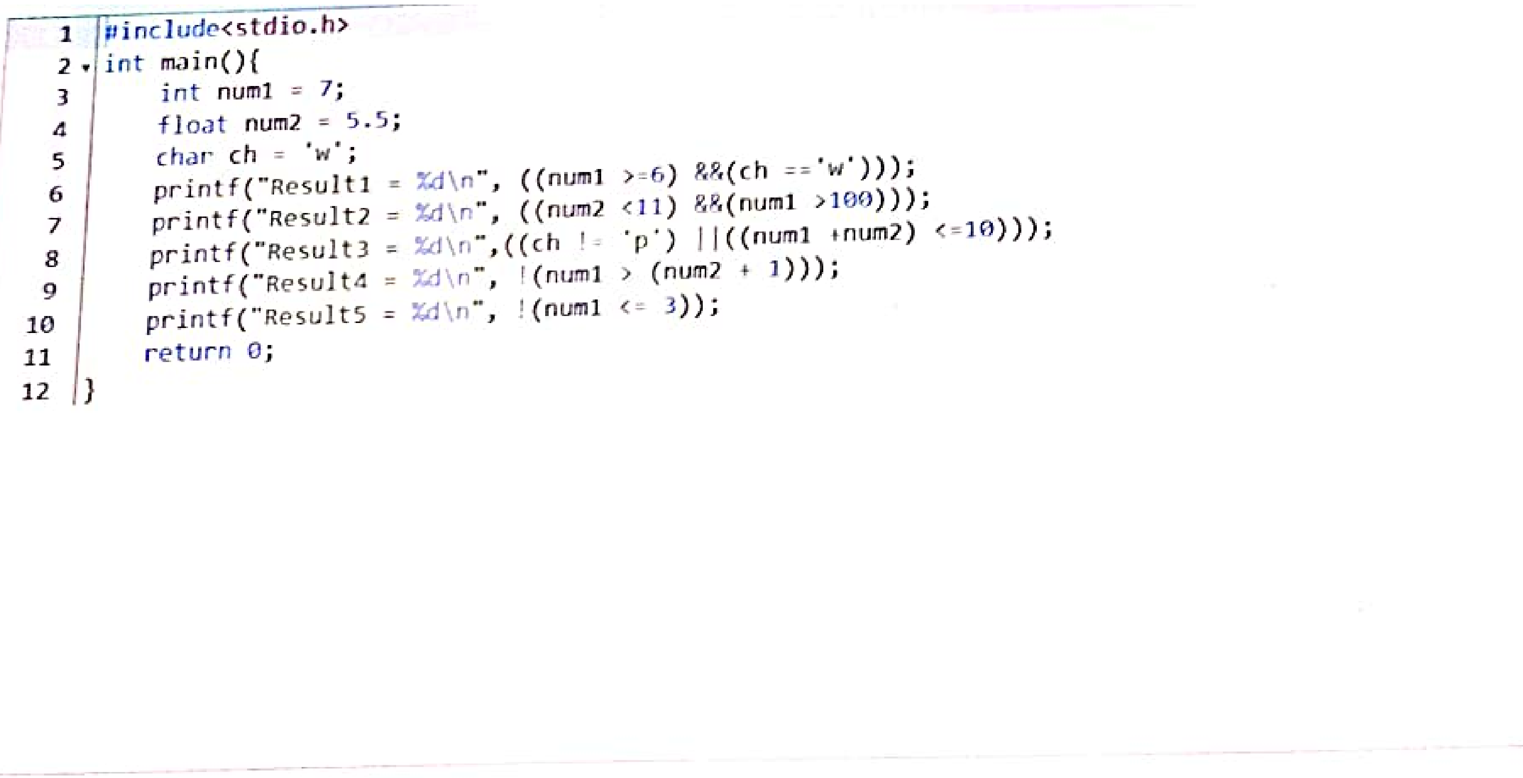
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int num1 = 7;
    float num2 = 5.5;
    char ch ='w';
    printf("Result1 = %d\n",(num1 > 5));
    printf("Result2 = %d\n",((num1 + num2) <= 10));
    printf("Result3 = %d\n",(ch == 119));
    printf("Result4 = %d\n",(ch !='p'));
    printf("Result5 = %d",(ch >=10*(num1 + num2)));
    return 0;
}
```

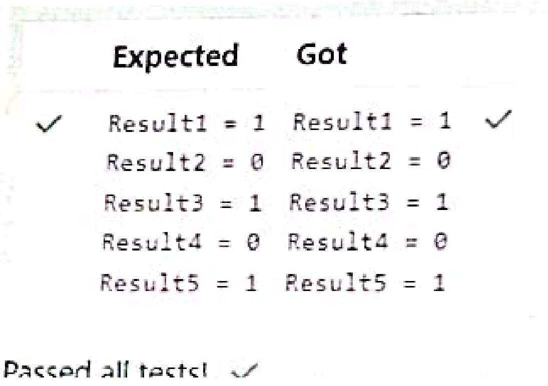
	Expected	Got	

✓	Result1 = 1
	Result2 = 0
	Result3 = 1
	Result4 = 1
	Result5 = 0
		Result1 = 1
		Result2 = 0
		Result3 = 1
		Result4 = 1
		Result5 = 0
			✓

Passed all tests! ✓



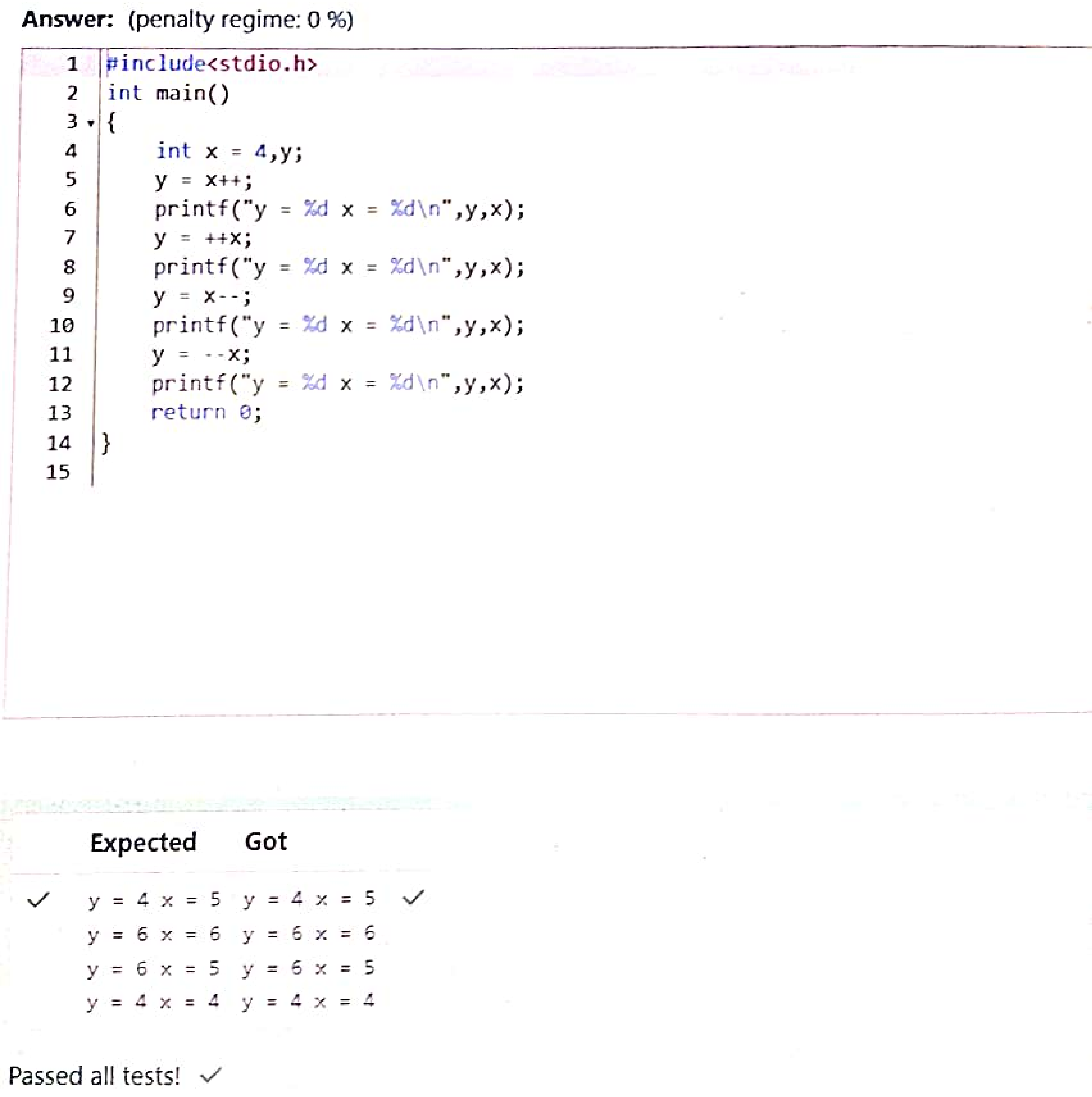
```c
#include<stdio.h>
int main(){
    int num1 = 7;
    float num2 = 5.5;
    char ch = 'w';
    printf("Result1 = %d\n", ((num1 >=6) &&(ch =='w')));
    printf("Result2 = %d\n", ((num2 <11) &&(num1 >100)));
    printf("Result3 = %d\n",((ch != 'p') ||((num1 +num2) <=10)));
    printf("Result4 = %d\n", !(num1 > (num2 + 1)));
    printf("Result5 = %d\n", !(num1 <= 3));
    return 0;
}
```



	Expected	Got	

✓	Result1 = 1
	Result2 = 0
	Result3 = 1
	Result4 = 0
	Result5 = 1
		Result1 = 1
		Result2 = 0
		Result3 = 1
		Result4 = 0
		Result5 = 1
			✓

Passed all tests! ✓



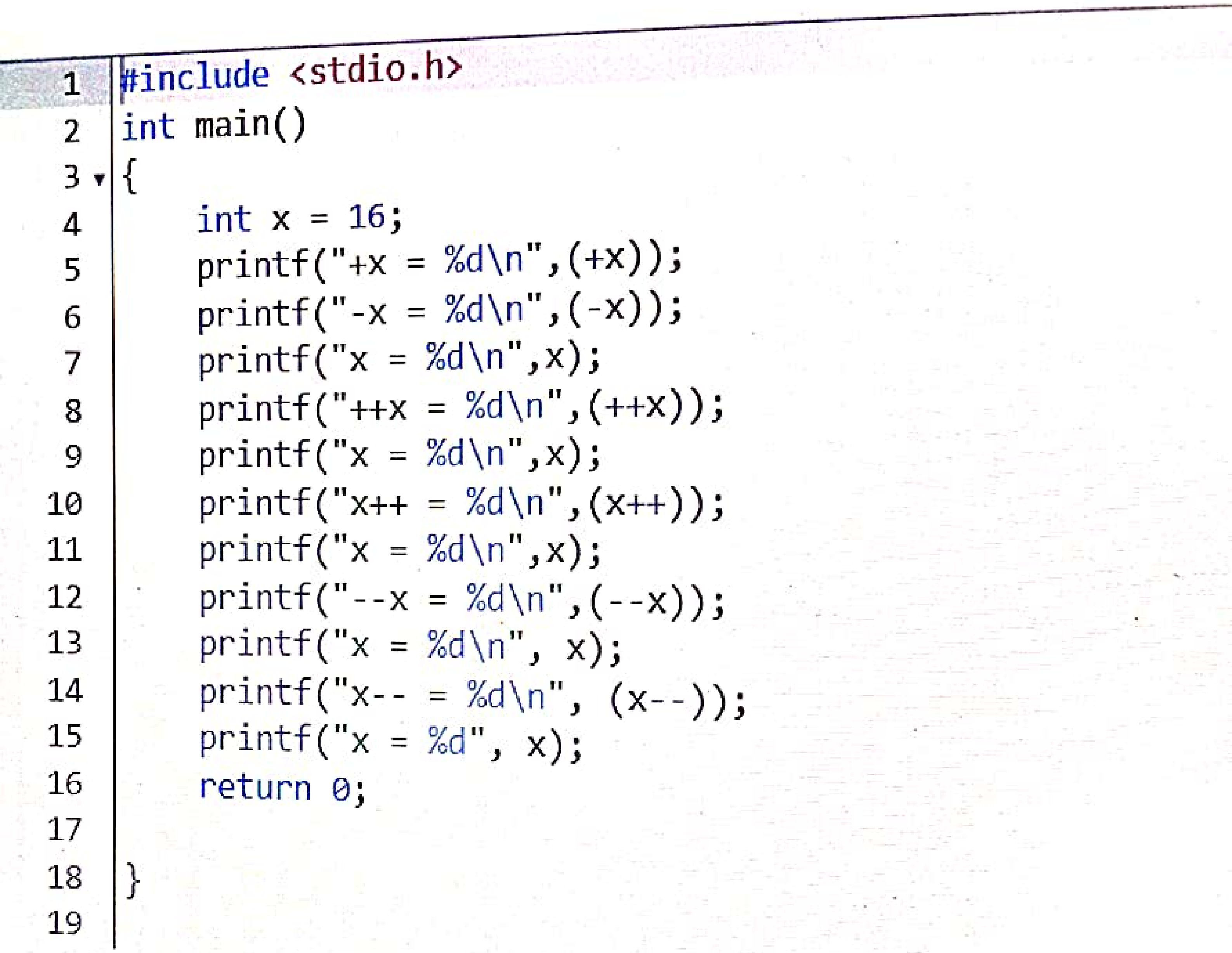
Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int x = 4,y;
    y = x++;
    printf("y = %d x = %d\n",y,x);
    y = ++x;
    printf("y = %d x = %d\n",y,x);
    y = x--;
    printf("y = %d x = %d\n",y,x);
    y = --x;
    printf("y = %d x = %d\n",y,x);
    return 0;
}

```

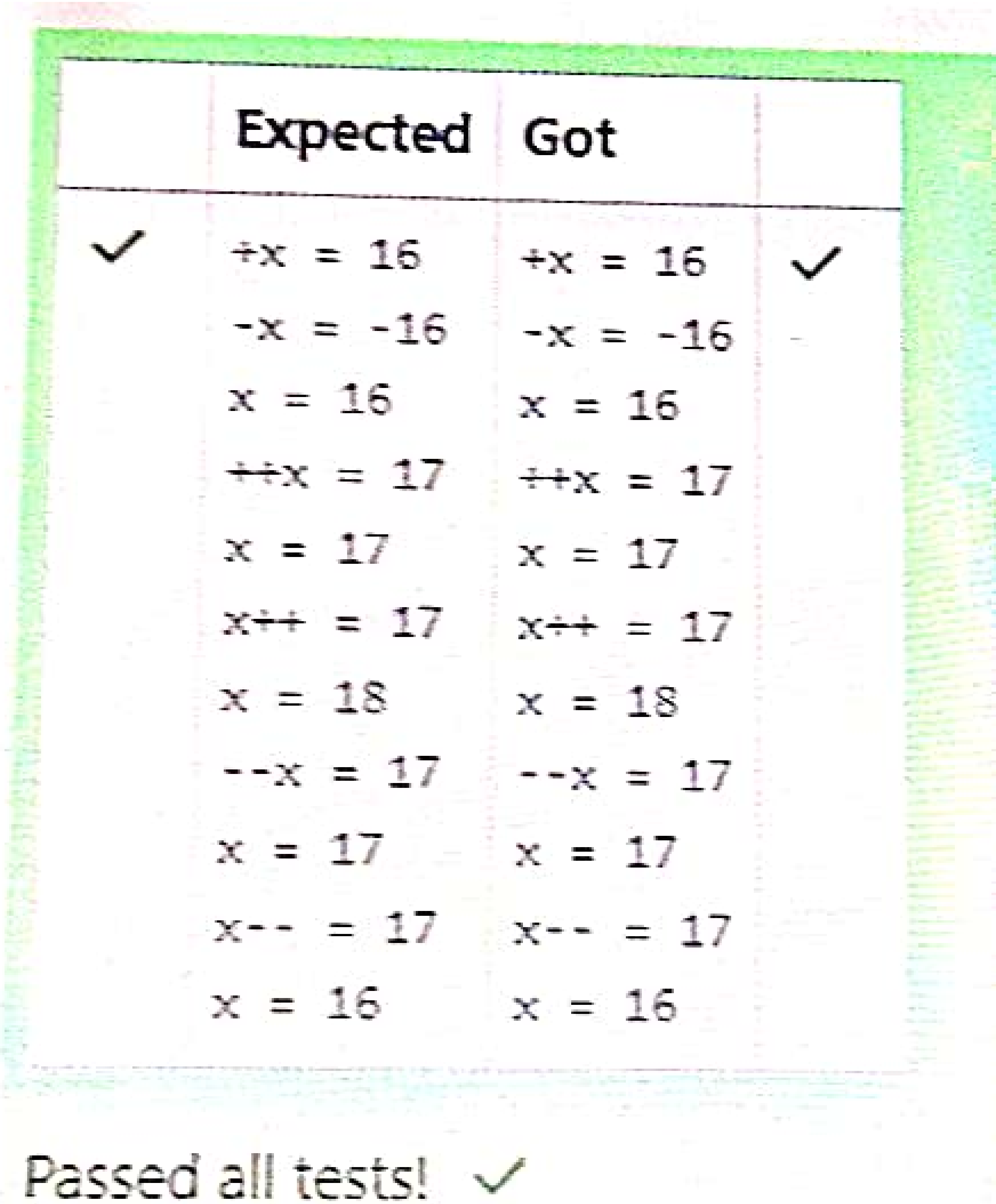
	Expected	Got	

✓	y = 4 x = 5
	y = 6 x = 6
	y = 6 x = 5
	y = 4 x = 4
		y = 4 x = 5
		y = 6 x = 6
		y = 6 x = 5
		y = 4 x = 4
			✓

Passed all tests! ✓



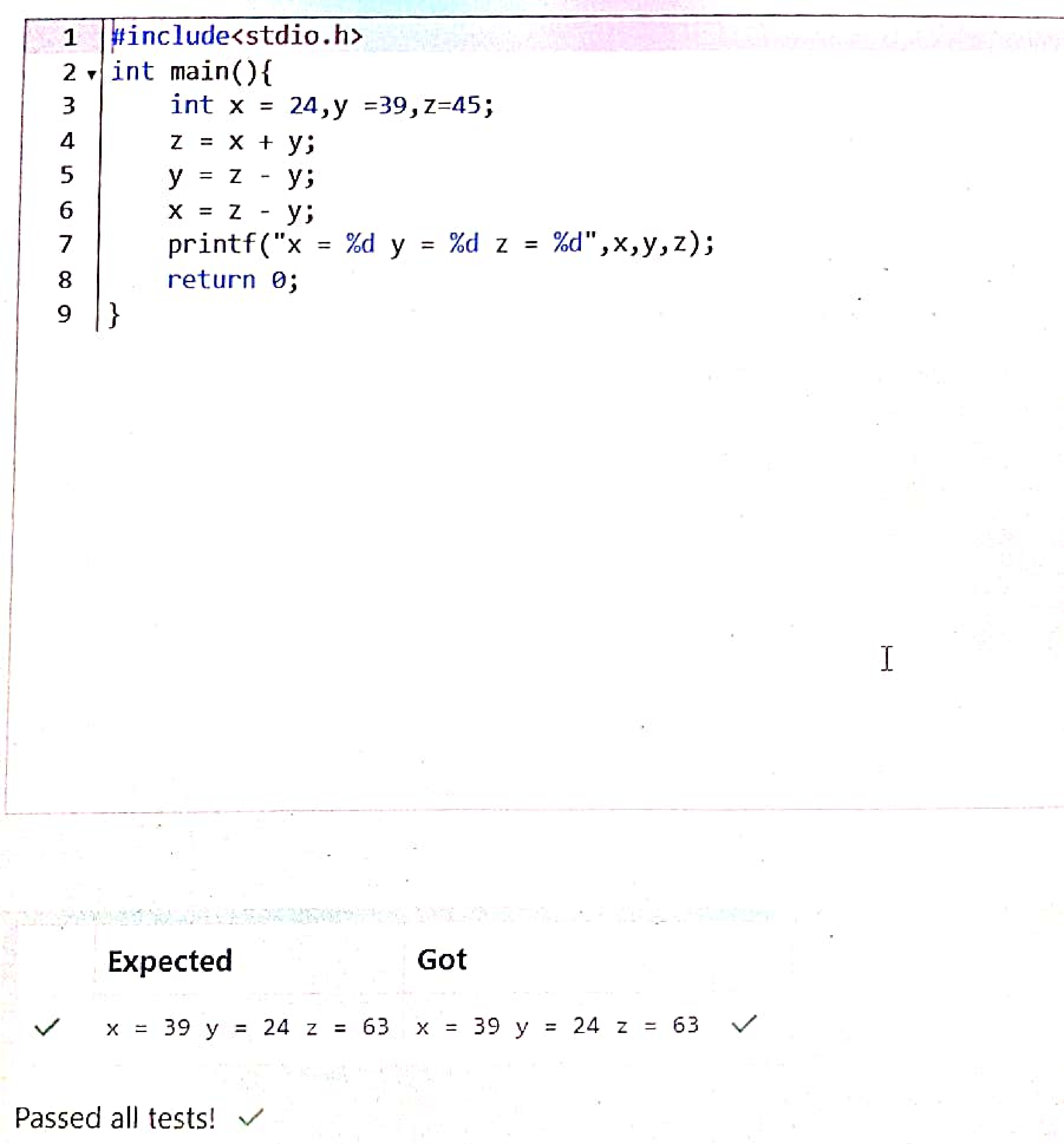
```
#include <stdio.h>
int main()
{
    int x = 16;
    printf("+x = %d\n",(+x));
    printf("-x = %d\n",(-x));
    printf("x = %d\n",x);
    printf("++x = %d\n",(++x));
    printf("x = %d\n",x);
    printf("x++ = %d\n",(x++));
    printf("x = %d\n",x);
    printf("--x = %d\n",(--x));
    printf("x = %d\n", x);
    printf("x-- = %d\n", (x--));
    printf("x = %d", x);
    return 0;

}
```



	Expected	Got	

✓	+x = 16
	-x = -16
	x = 16
	++x = 17
	x = 17
	x++ = 17
	x = 18
	--x = 17
	x = 17
	x-- = 17
	x = 16
		+x = 16
		-x = -16
		x = 16
		++x = 17
		x = 17
		x++ = 17
		x = 18
		--x = 17
		x = 17
		x-- = 17
		x = 16
			✓

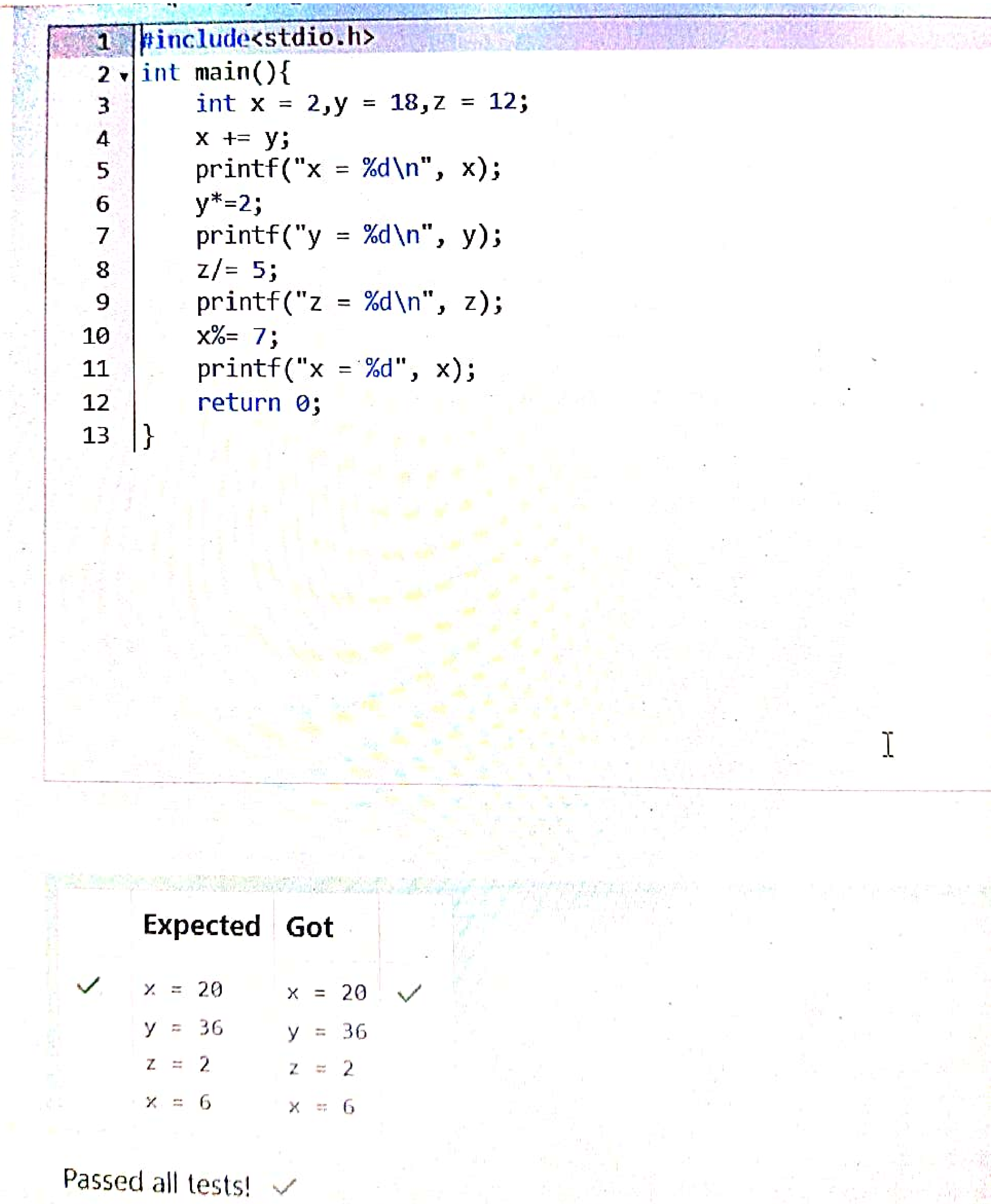
Passed all tests! ✓



```c
#include<stdio.h>
int main(){
    int x = 24,y =39,z=45;
    z = x + y;
    y = z - y;
    x = z - y;
    printf("x = %d y = %d z = %d",x,y,z);
    return 0;
}
```

	Expected	Got	
✓	x = 39 y = 24 z = 63	x = 39 y = 24 z = 63	✓

Passed all tests! ✓

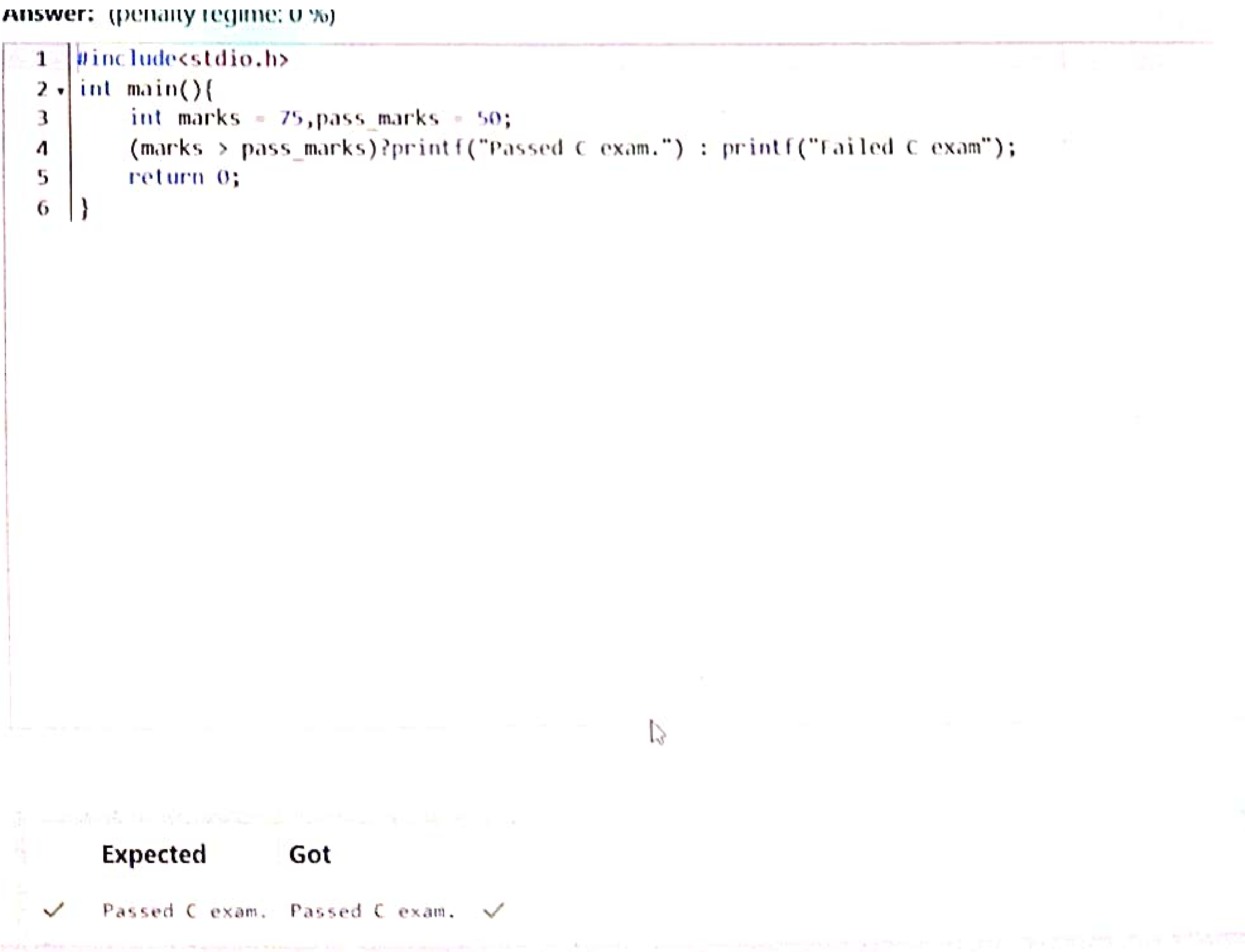


```c
#include<stdio.h>
int main(){
    int x = 2,y = 18,z = 12;
    x += y;
    printf("x = %d\n", x);
    y*=2;
    printf("y = %d\n", y);
    z/= 5;
    printf("z = %d\n", z);
    x%= 7;
    printf("x = %d", x);
    return 0;
}
```

	Expected	Got	

✓	x = 20
	y = 36
	z = 2
	x = 6
		x = 20
		y = 36
		z = 2
		x = 6
			✓

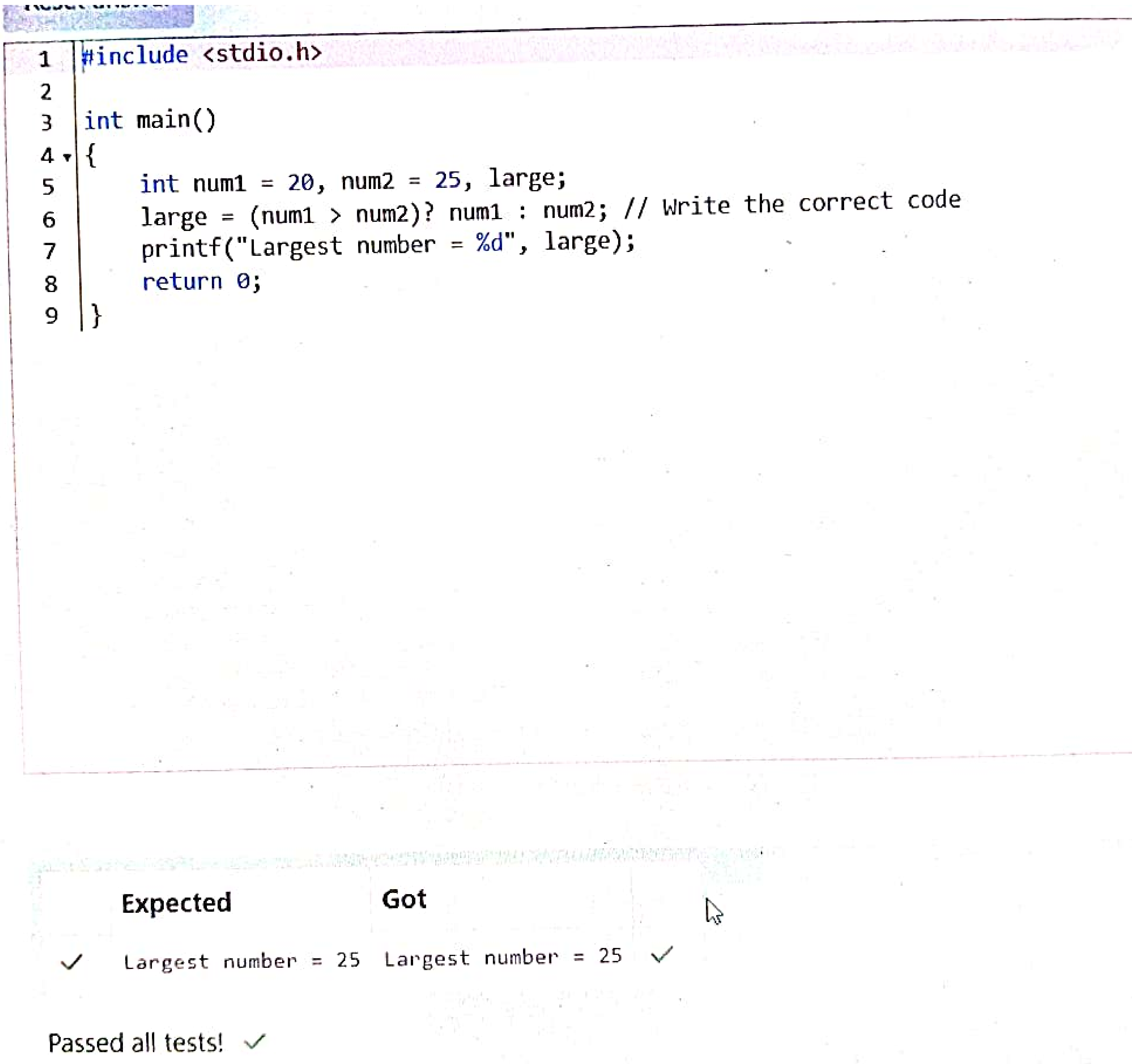
Passed all tests! ✓



Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int marks = 75,pass_marks = 50;
    (marks > pass_marks)?printf("Passed C exam.") : printf("Failed C exam");
    return 0;
}
```

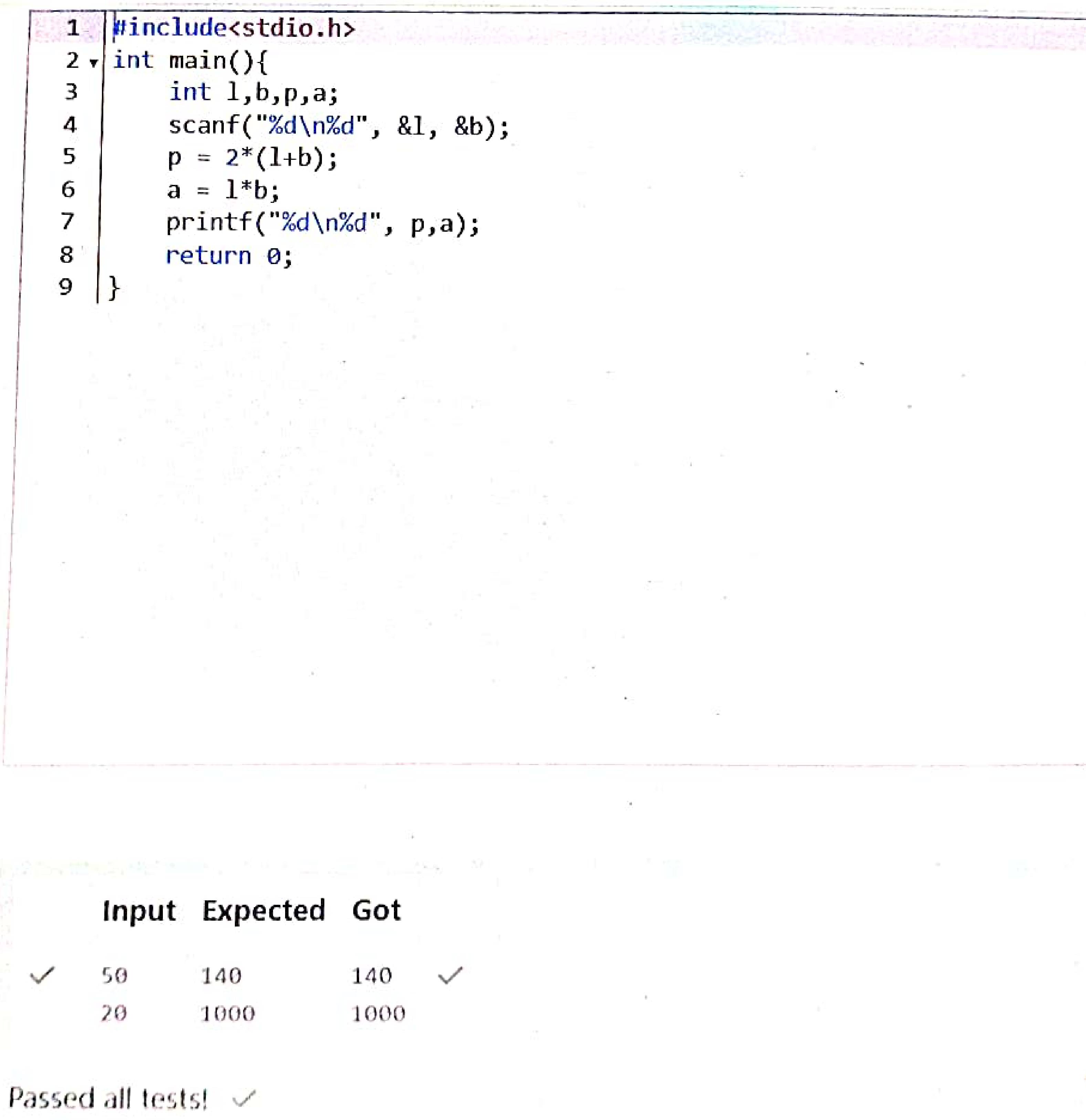
	Expected	Got	
✓	Passed C exam.	Passed C exam.	✓



```c
#include <stdio.h>

int main()
{
    int num1 = 20, num2 = 25, large;
    large = (num1 > num2)? num1 : num2; // Write the correct code
    printf("Largest number = %d", large);
    return 0;
}
```

	Expected	Got	
✓	Largest number = 25	Largest number = 25	✓

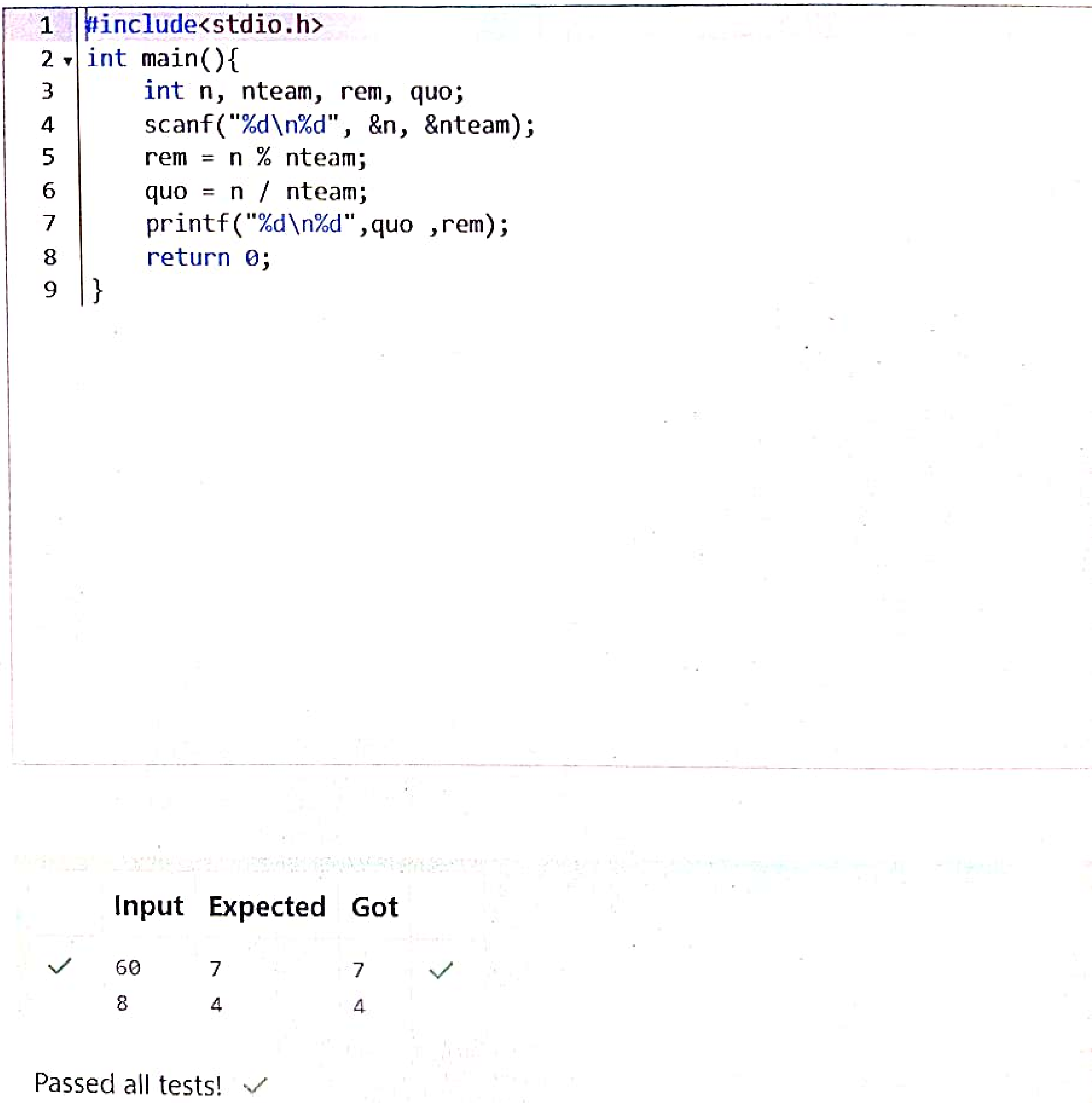
Passed all tests! ✓



```c
#include<stdio.h>
int main(){
    int l,b,p,a;
    scanf("%d\n%d", &l, &b);
    p = 2*(l+b);
    a = l*b;
    printf("%d\n%d", p,a);
    return 0;
}
```

	Input	Expected	Got	
✓	50 20	140 1000	140 1000	✓

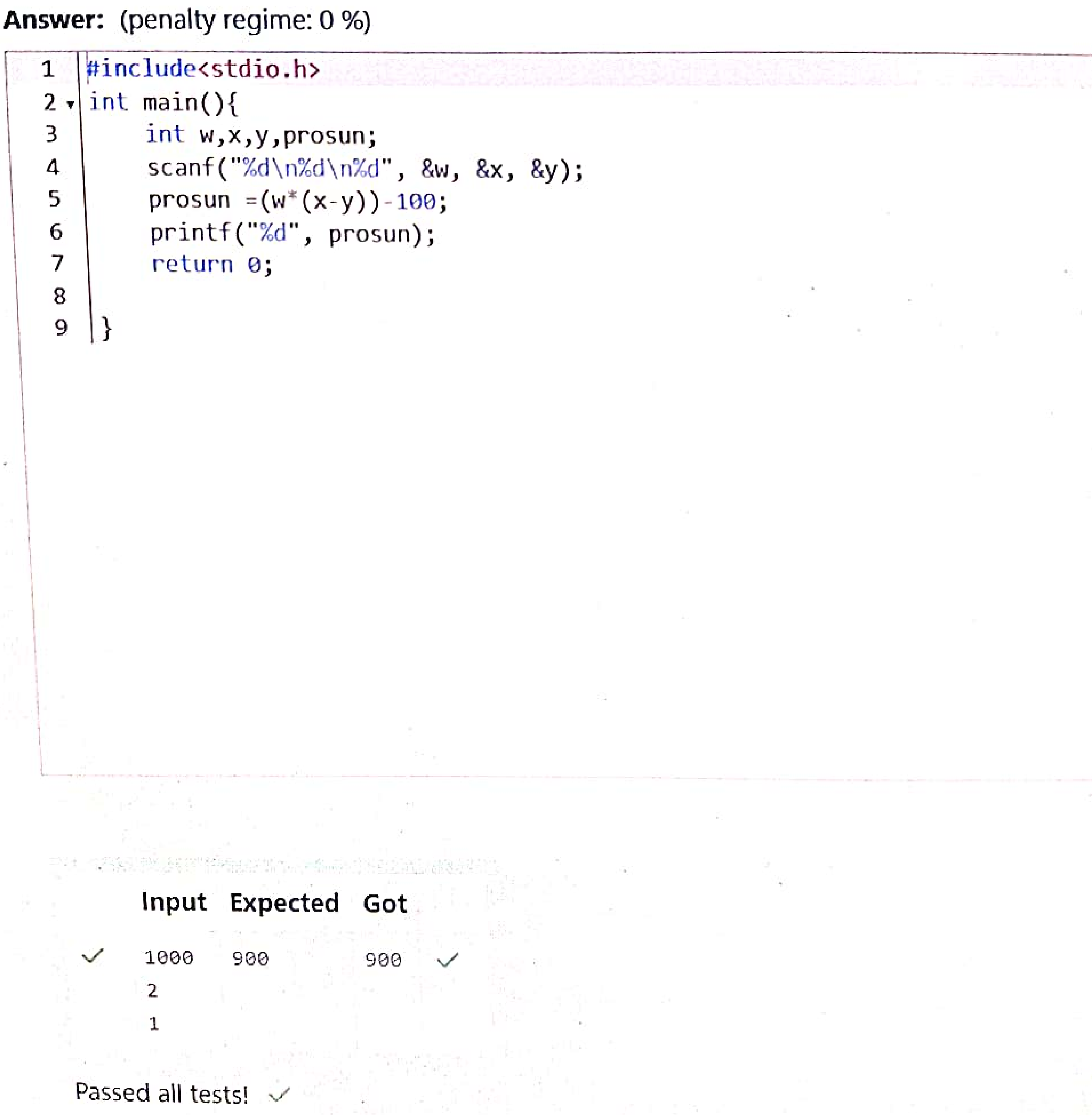
Passed all tests! ✓



```c
#include<stdio.h>
int main(){
    int n, nteam, rem, quo;
    scanf("%d\n%d", &n, &nteam);
    rem = n % nteam;
    quo = n / nteam;
    printf("%d\n%d",quo ,rem);
    return 0;
}
```

	Input	Expected	Got	
✓	60 8	7 4	7 4	✓

Passed all tests! ✓

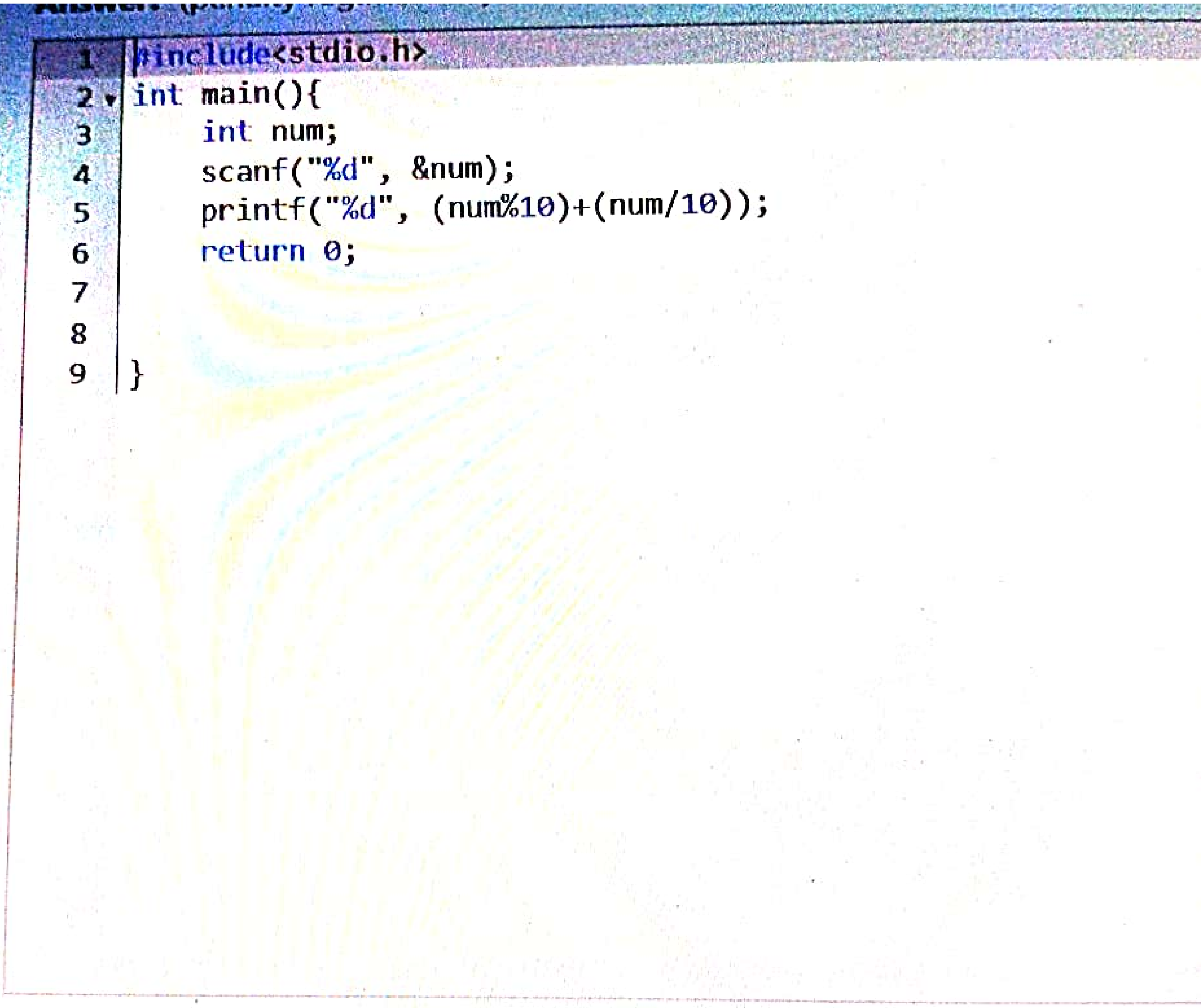


Answer: (penalty regime: 0 %)

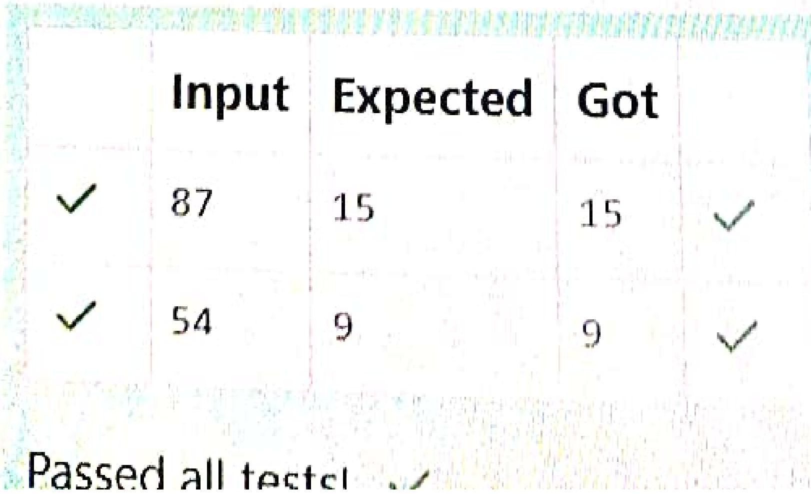
```c
#include<stdio.h>
int main(){
    int w,x,y,prosun;
    scanf("%d\n%d\n%d", &w, &x, &y);
    prosun =(w*(x-y))-100;
    printf("%d", prosun);
    return 0;

}
```

	Input	Expected	Got	
✓	1000 2 1	900	900	✓

Passed all tests! ✓

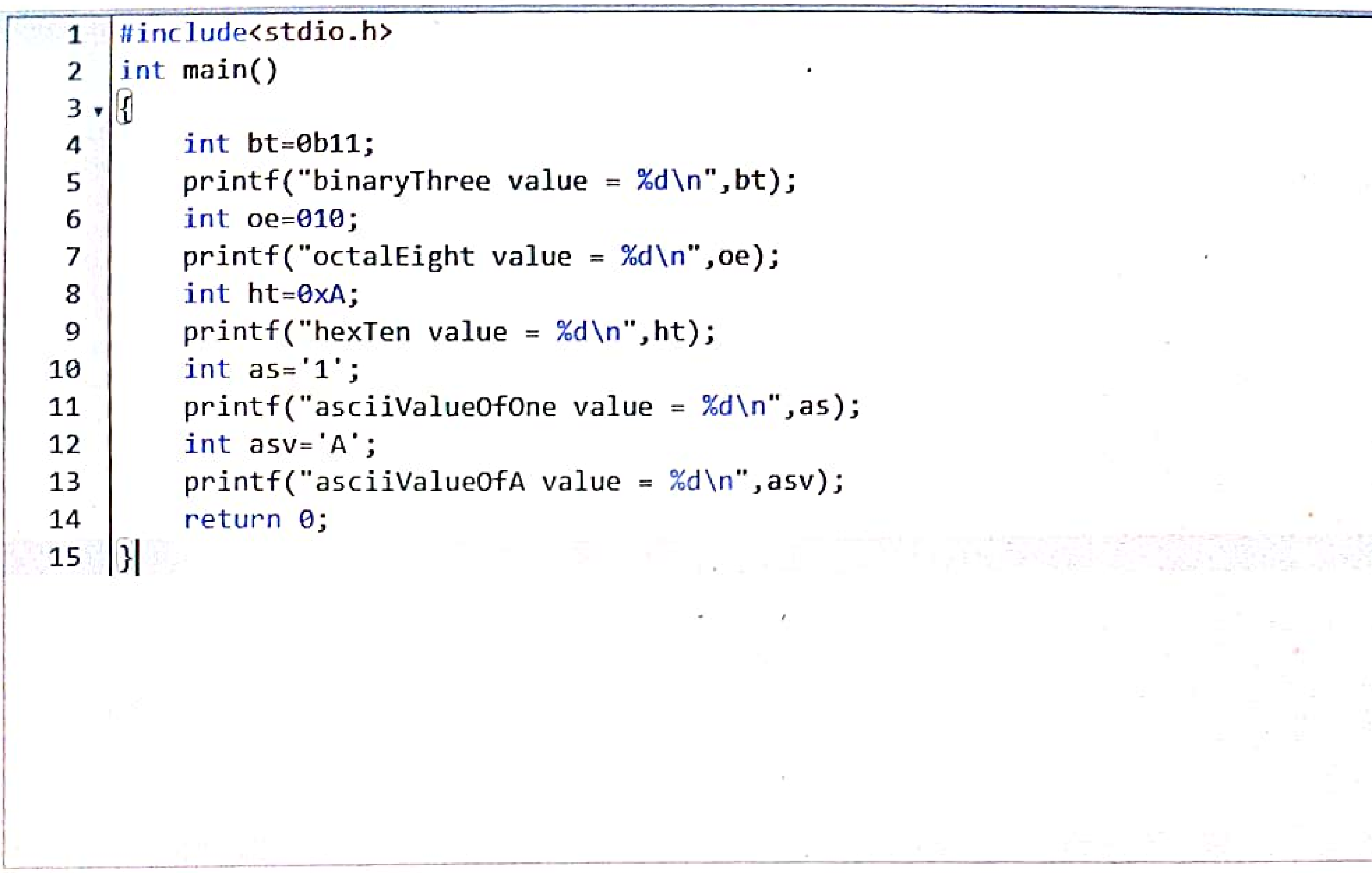


```c
#include<stdio.h>
int main(){
    int num;
    scanf("%d", &num);
    printf("%d", (num%10)+(num/10));
    return 0;


}
```

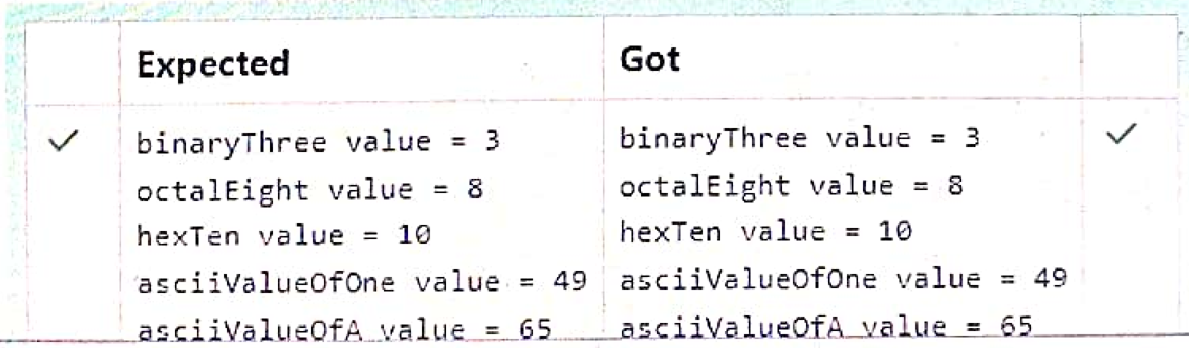


	Input	Expected	Got	
✓	87	15	15	✓
✓	54	9	9	✓

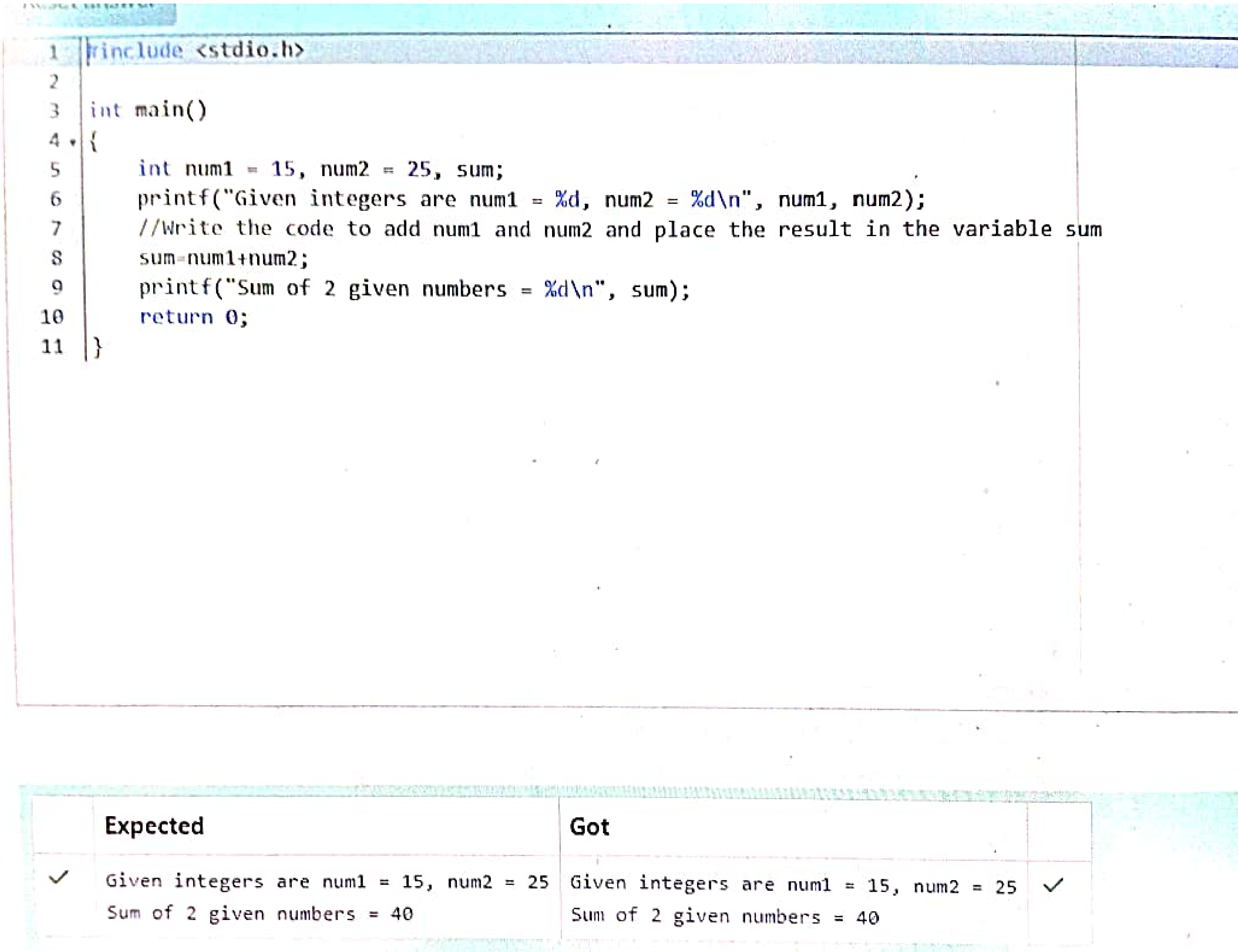
Passed all tests! ✓



```c
#include<stdio.h>
int main()
{
    int bt=0b11;
    printf("binaryThree value = %d\n",bt);
    int oe=010;
    printf("octalEight value = %d\n",oe);
    int ht=0xA;
    printf("hexTen value = %d\n",ht);
    int as='1';
    printf("asciiValueOfOne value = %d\n",as);
    int asv='A';
    printf("asciiValueOfA value = %d\n",asv);
    return 0;
}
```



	Expected	Got	
✓	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	✓



```c
#include <stdio.h>

int main()
{
    int num1 = 15, num2 = 25, sum;
    printf("Given integers are num1 = %d, num2 = %d\n", num1, num2);
    //Write the code to add num1 and num2 and place the result in the variable sum
    sum=num1+num2;
    printf("Sum of 2 given numbers = %d\n", sum);
    return 0;
}
```

	Expected	Got	
✓	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	✓