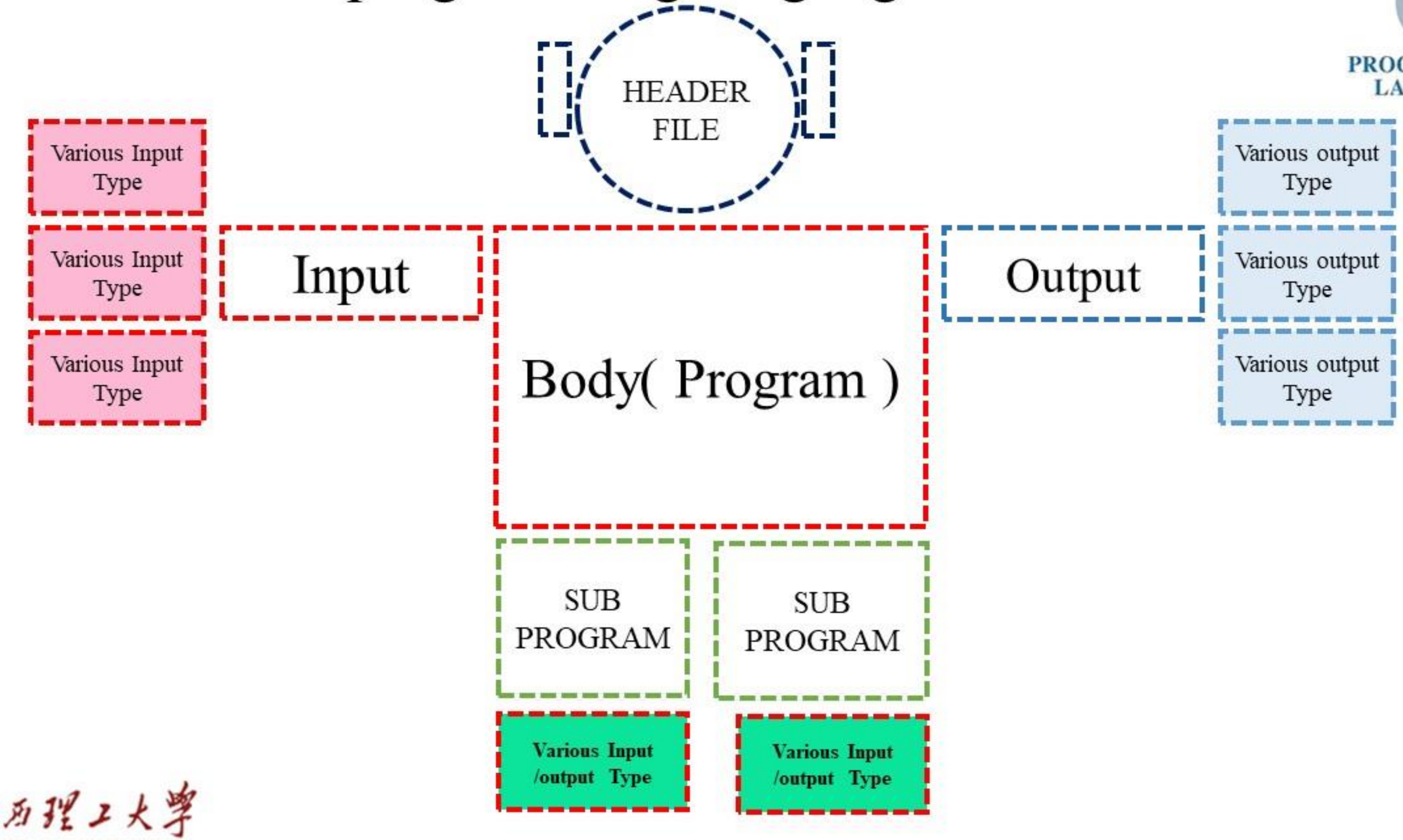
Jiangxi University of Science and Technology

Chapter 3 Processing and Interactive Input

Leture0303 Interactive Input and Formatted Output_section 2

My idea about all programming language

THE C PROGRAMMING LANGUAGE



HEADER FILE

Various Input Type

Various Input Type

Various Input Type

Input

Body(Program)

Output

Various output Type

Various output Type

Various output Type

SUB PROGRAM

SUB PROGRAM

Various Input /output Type

Various Input /output Type

江西理工大學

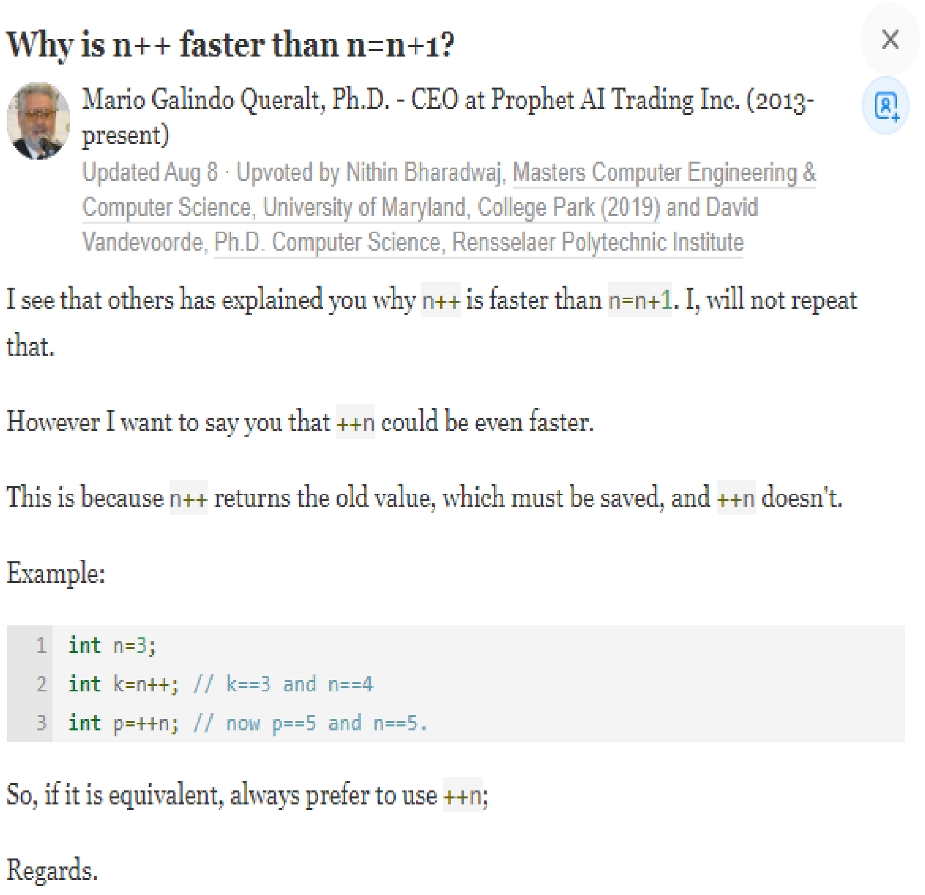
First answer this question

Why is n++ faster than n=n+1?

However I want to say you that ++n could be even faster.
This is because n++ returns the old value, which must be saved, and ++n doesn't.
Example:

```
1.int n=3;
2.int k=n++; // k==3 and n==4
3.int p=++n; // now p==5 and n==5.
```

So, if it is equivalent, always prefer to use ++n;



Why is n++ faster than n=n+1?

Mario Galindo Queralt, Ph.D. - CEO at Prophet AI Trading Inc. (2013-present)

Updated Aug 8 · Upvoted by Nithin Bharadwaj, Masters Computer Engineering & Computer Science, University of Maryland, College Park (2019) and David Vandevoorde, Ph.D. Computer Science, Rensselaer Polytechnic Institute

I see that others has explained you why n++ is faster than n=n+1. I, will not repeat that.

However I want to say you that ++n could be even faster.

This is because n++ returns the old value, which must be saved, and ++n doesn't.

Example:

```
int n=3;
int k=n++; // k==3 and n==4
int p=++n; // now p==5 and n==5.
```

So, if it is equivalent, always prefer to use ++n;

Regards.

Task write a smple code and compare

3.4 Formatted Output

- Other Number Bases

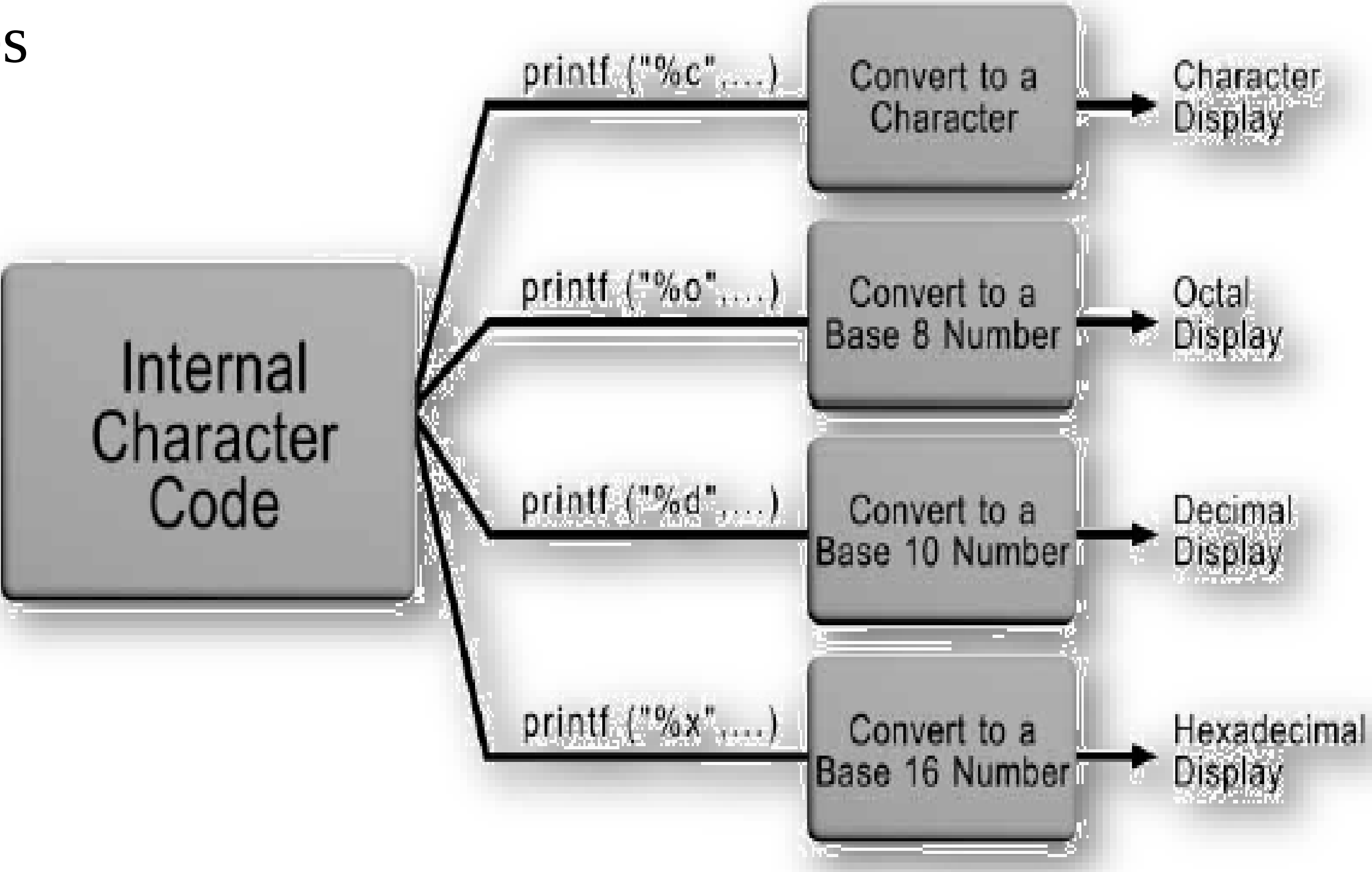


Figure 3.8 Character display options

JUST remember

- %d (print as a decimal integer)
- %6d (print as a decimal integer with a width of at least 6 wide)
- %06d (print as a decimal integer with a width of at least 6 wide,precedes with 0)
- %f (print as a floating point)
- %4f (print as a floating point with a width of at least 4 wide)
- %.4f (print as a floating point with a precision of four characters after the decimal point)
- %3.2f (print as a floating point at least 3 wide and a precision of 2)

JUST remember

INPUT		OUTPUT:
int a=10;	printf(“%d”,a);	10
	printf(“%4d”,a);	10 (represents white space in the output, for clarity convenience I used)
	printf(“%04d”,a);	0010

INPUT		OUTPUT:
float f=1.2;	printf(“%f”,f);	1.200000
	printf(“%9f”,f);	1.200000 (represents white space in the output, for clarity convenience I used _)
	printf(“%.2f”,f);	1.20
	printf(“%6.2f”,f);	1.20

JUST remember

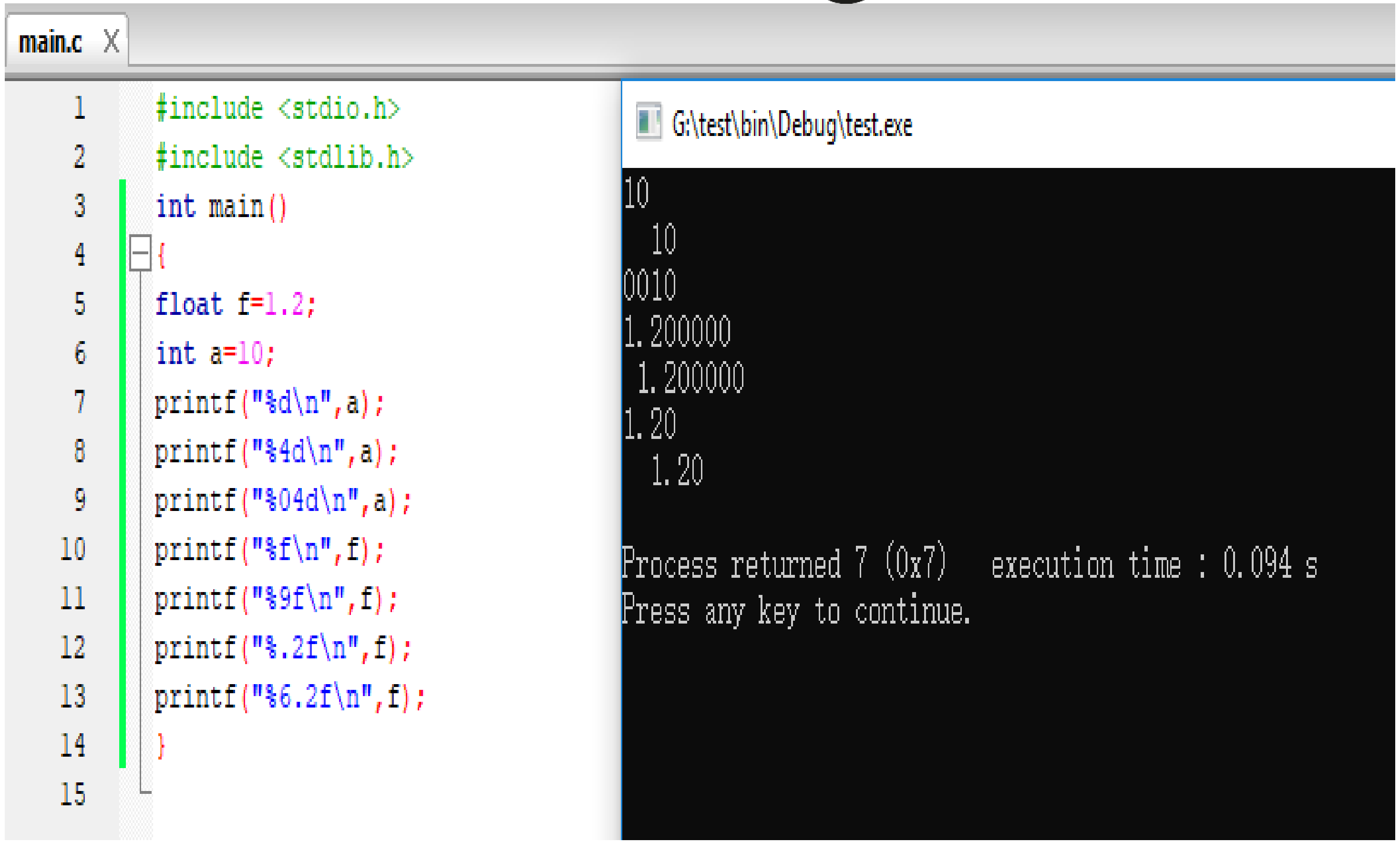
	OUTPUT:
%3.2f	//(print as a floating point at least 3 wide and a precision of 2)
%0.2lf	//(print as a floating point at least 0 wide and a precision of 2)
%.2lf	//(print as a floating point at least 0(default) wide and a precision of 2)

	OUTPUT:
"%0.2lf", 0.123	0.12 (zero padded min. width of 0, 2 decimal places).
"%6.2lf", 0.123	0.12 (space padded min. width of 6, 2 decimal places).
"%0.6lf", 0.123	0.123000 (min width of 0, 6 decimal places).
"%06.2lf", 0.123	000.12 (zero padded min. width of 6, 2 decimal places).

NOW test this

THE C PROGRAMMING LANGUAGE

```
#include <stdio.h>
int main()
{
float f=1.2;
int a=10;
printf("%d\n",a);
printf("%4d\n",a);
printf("%04d\n",a);
printf("%f\n",f);
printf("%9f\n",f);
printf("%.2f\n",f);
printf("%6.2f\n",f);
}
```



```
#include <stdio.h>
#include <stdlib.h>
int main()
{
float f=1.2;
int a=10;
printf("%d\n",a);
printf("%4d\n",a);
printf("%04d\n",a);
printf("%f\n",f);
printf("%9f\n",f);
printf("%.2f\n",f);
printf("%6.2f\n",f);
}
```

```
10
  10
0010
1.200000
 1.200000
1.20
  1.20

Process returned 7 (0x7)   execution time : 0.094 s
Press any key to continue.
```

Jiangxi University of Science and Technology

Chapter 3 Processing and Interactive Input

Leture0302 Mathematical Library Functions

3.2 Mathematical Library Functions

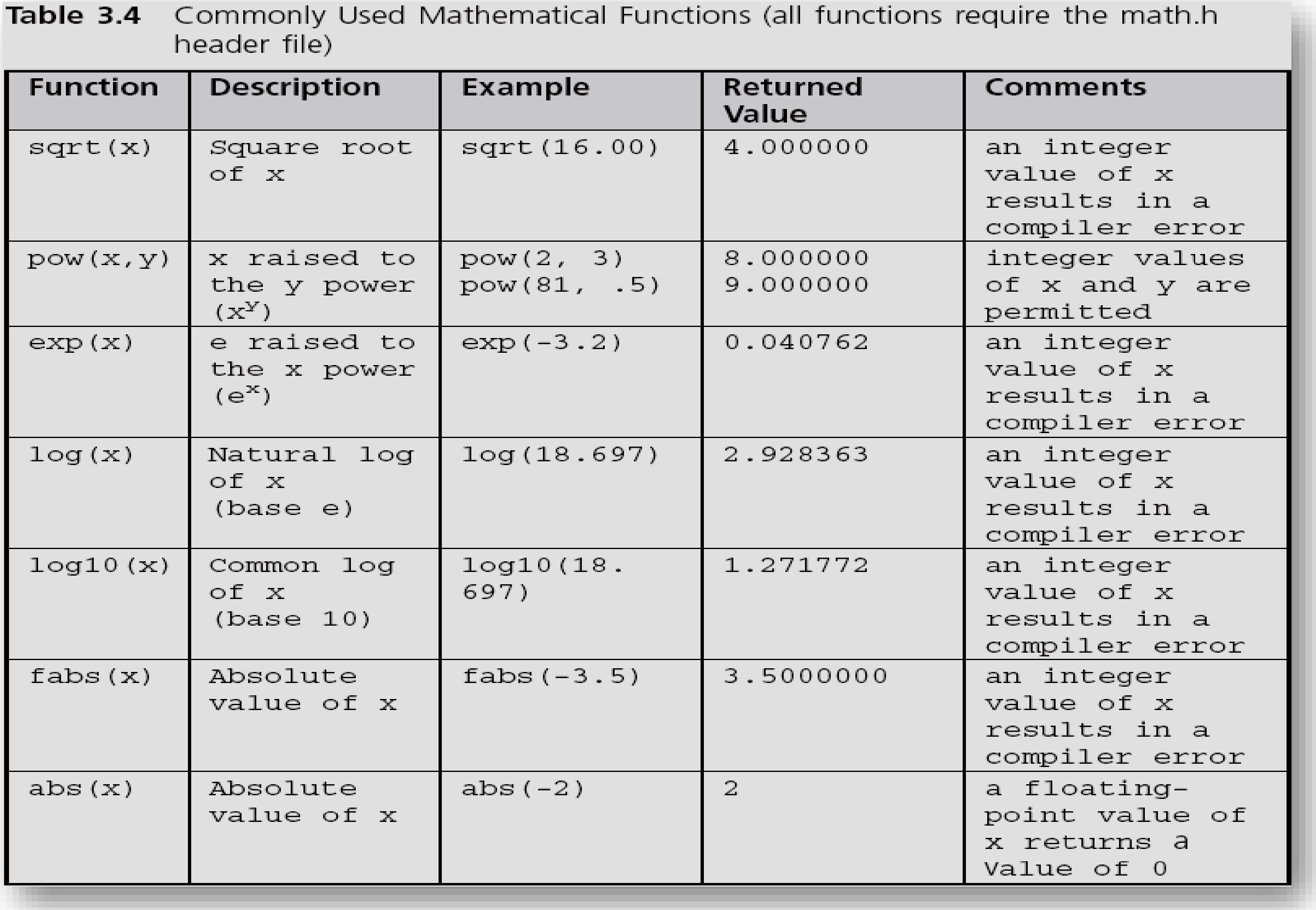


Table 3.4 Commonly Used Mathematical Functions (all functions require the math.h header file)

Function	Description	Example	Returned Value	Comments
sqrt(x)	Square root of x	sqrt(16.00)	4.000000	an integer value of x results in a compiler error

pow(x,y)	x raised to the y power (x^y)	pow(2, 3)
		pow(81, .5)
			8.000000
			9.000000
				integer values of x and y are permitted

exp(x)	e raised to the x power (e^x)	exp(-3.2)	0.040762	an integer value of x results in a compiler error
log(x)	Natural log of x (base e)	log(18.697)	2.928363	an integer value of x results in a compiler error
log10(x)	Common log of x (base 10)	log10(18.697)	1.271772	an integer value of x results in a compiler error
fabs(x)	Absolute value of x	fabs(-3.5)	3.5000000	an integer value of x results in a compiler error
abs(x)	Absolute value of x	abs(-2)	2	a floating-point value of x returns a Value of 0

3.2 Mathematical Library Functions

- **float sqrt(float x)**
  - The argument to sqrt must be floating-point value;
  - Return value is floating-point value;
  - passing an integer value results in a compiler error
  - must include ***#include <math.h>***

3.2 Mathematical Library Functions

- **float sqrt(float x)**

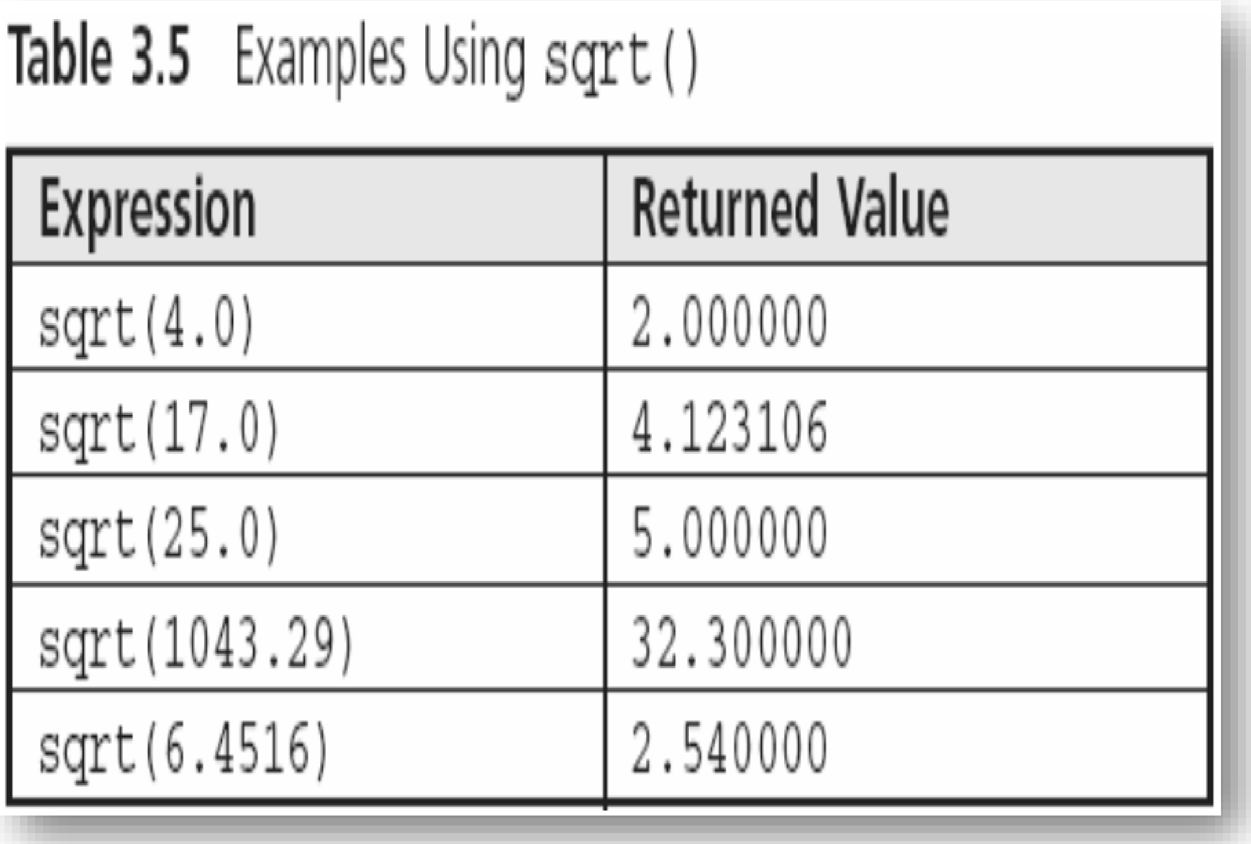


Table 3.5 Examples Using sqrt()

Expression	Returned Value
sqrt(4.0)	2.000000
sqrt(17.0)	4.123106
sqrt(25.0)	5.000000
sqrt(1043.29)	32.300000
sqrt(6.4516)	2.540000

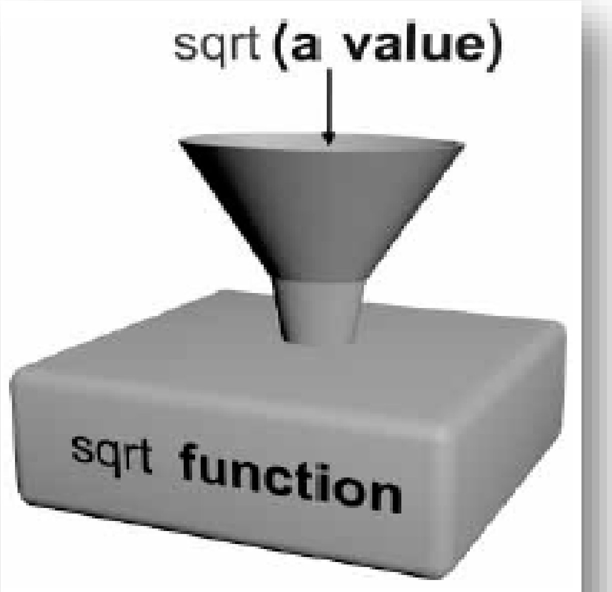


Figure 3.3 Passing data to the `sqrt()` function

3.2 Mathematical Library Functions

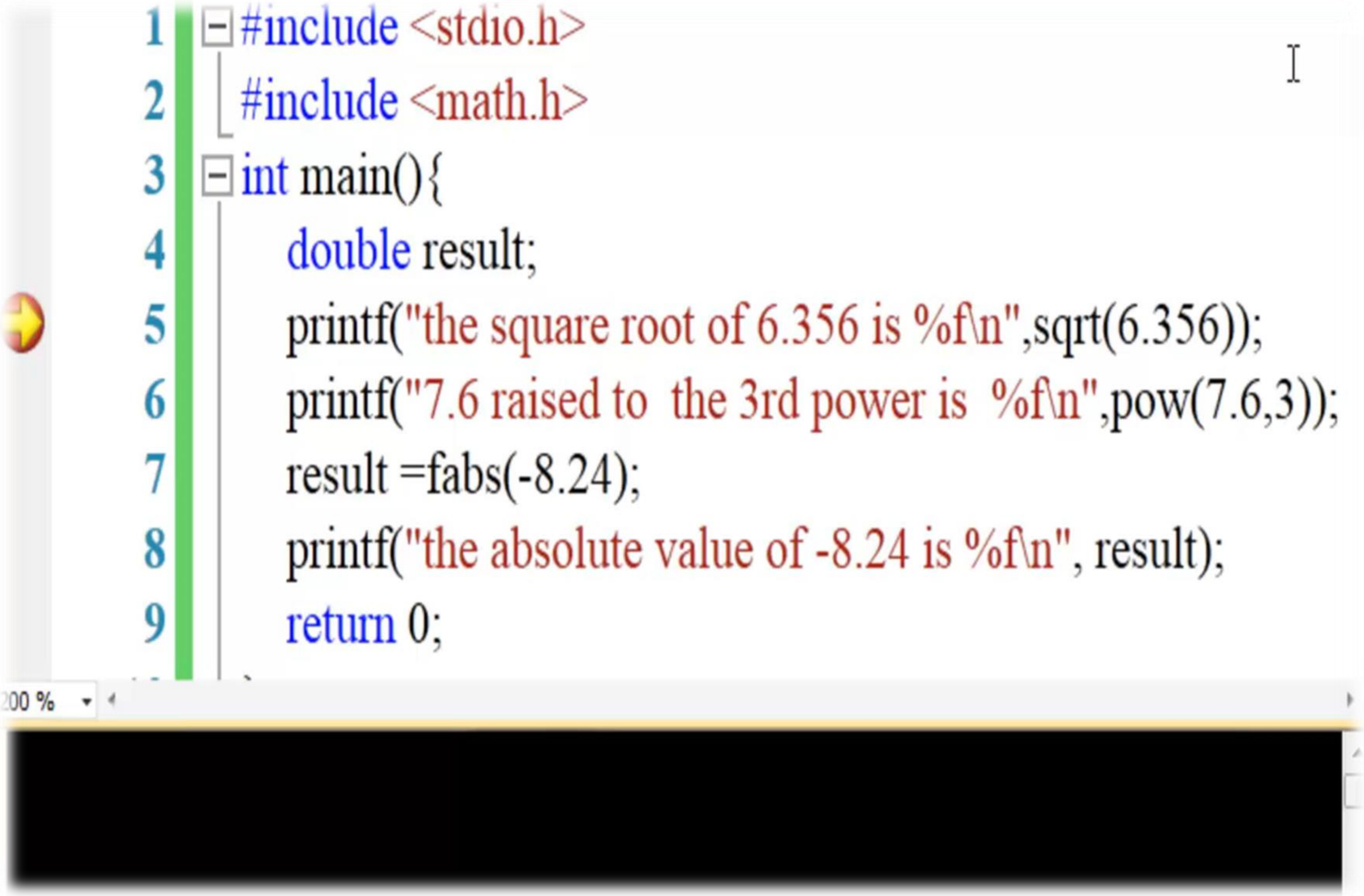
- Program 3.5

TEST Me!

```c
#include <stdio.h>
#include <math.h>
int main(){
        double result;
        printf("the square root of 6.356 is %f\n", sqrt(6.356));
        printf("7.6 raised to  the 3rd power is  %f\n", pow(7.6,3));
        result = fabs(-8.24);
        printf("the absolute value of -8.24 is %f\n", result);
        return 0;
}
```

3.2 Mathematical Library Functions

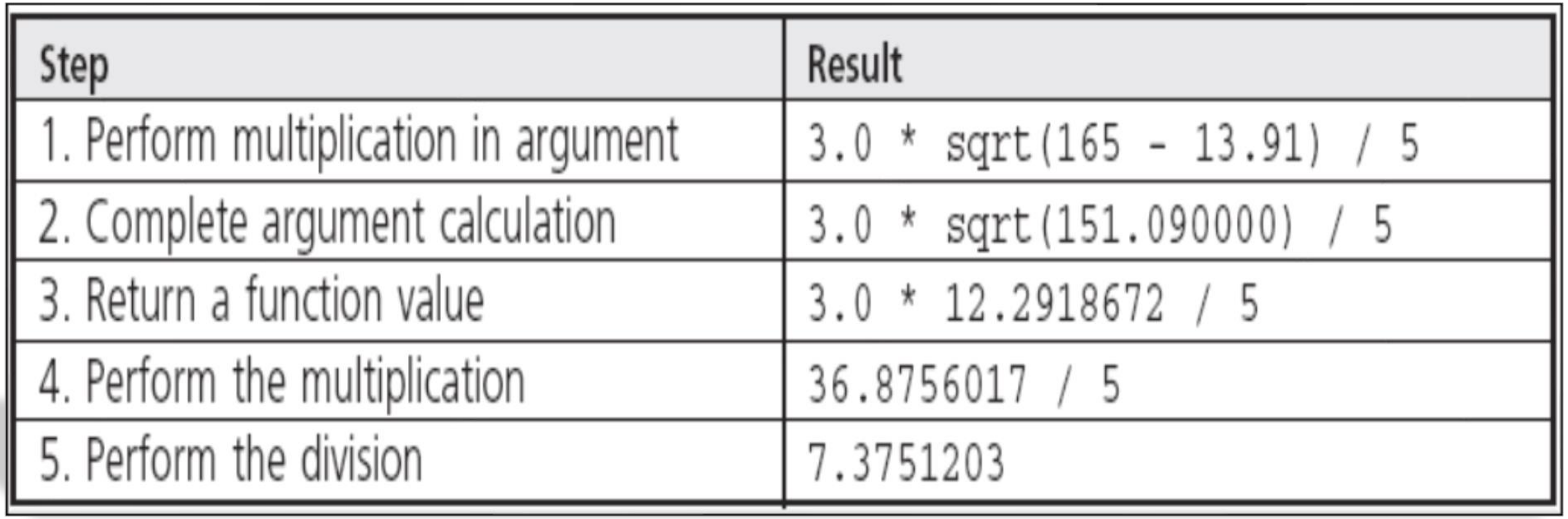
Program 3.5



```c
#include <stdio.h>
#include <math.h>
int main(){
    double result;
    printf("the square root of 6.356 is %f\n",sqrt(6.356));
    printf("7.6 raised to  the 3rd power is  %f\n",pow(7.6,3));
    result =fabs(-8.24);
    printf("the absolute value of -8.24 is %f\n", result);
    return 0;
```

3.2 Mathematical Library Functions

- The step-by-step evaluation of the expression
  - 3.0 * sqrt (5 * 33 - 13.91) / 5



Step	Result
1. Perform multiplication in argument	3.0 * sqrt(165 - 13.91) / 5
2. Complete argument calculation	3.0 * sqrt(151.090000) / 5
3. Return a function value	3.0 * 12.2918672 / 5
4. Perform the multiplication	36.8756017 / 5
5. Perform the division	7.3751203

3.2 Mathematical Library Functions

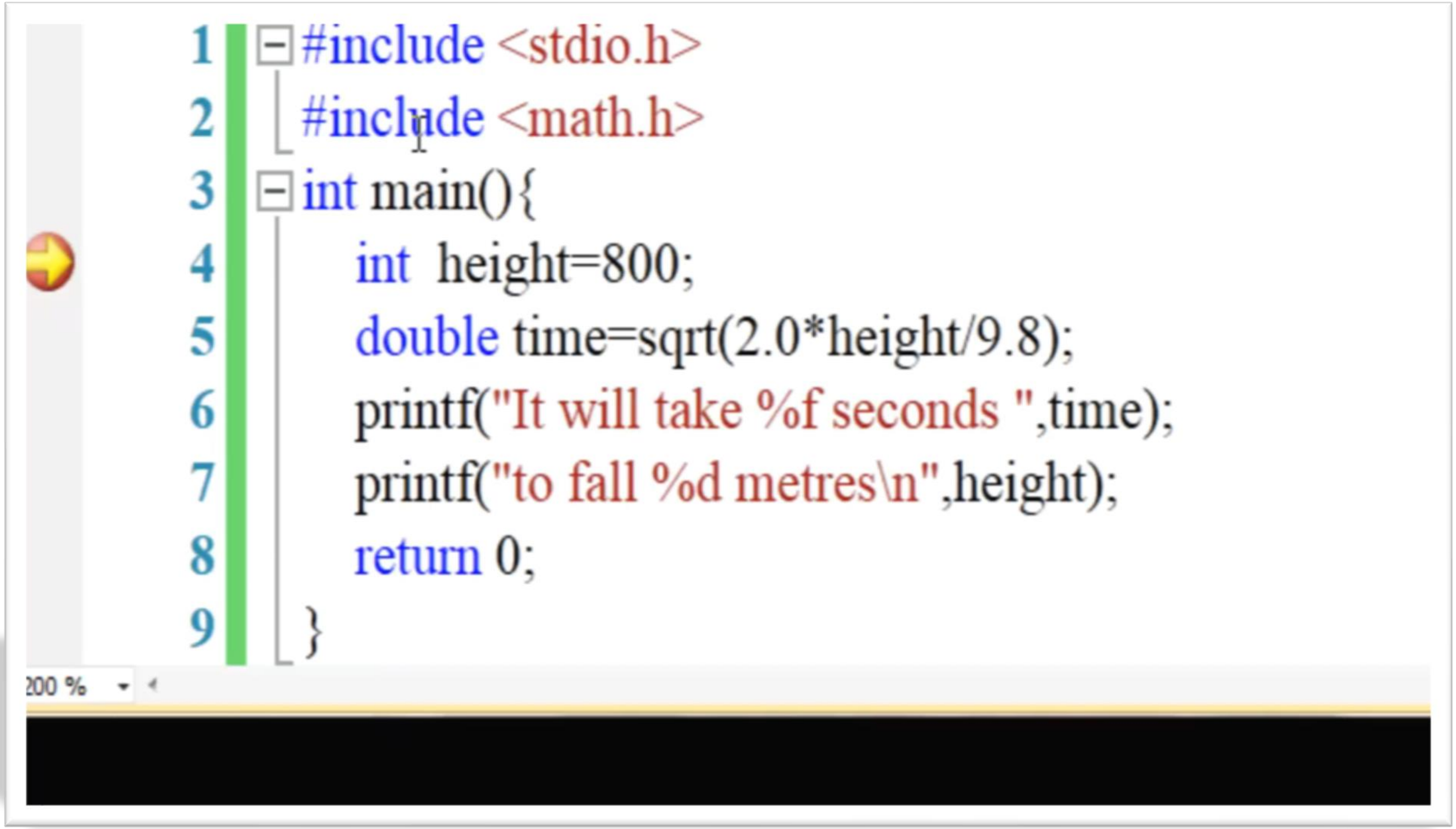
- Determine the time it takes a ball to hit the ground after it has been dropped from an 800-metre tower
  - time = sqrt(2 * height/g), where g = 9.8 metre/sec^2

```
#include <stdio.h>
#include <math.h>
int main(){
        int  height=800;
        double time=sqrt(2.0*height/9.8);
        printf("It will take %f seconds ",time);
        printf("to fall %d metres\n",height);
        return 0;
}
```

TEST Me!

3.2 Mathematical Library Functions

- Program 3.6



```
#include <stdio.h>
#include <math.h>
int main(){
    int  height=800;
    double time=sqrt(2.0*height/9.8);
    printf("It will take %f seconds ",time);
    printf("to fall %d metres\n",height);
    return 0;
}
```

Reference

- BOOK
- Some part of this PPT given by Prof 欧 **Chengtian Ouyang)**
- with special thank
- https://www.codingunit.com/c-tutorial hello-world