Computer Languages and Lab

Constants, Variables, and Data Types

Prof. Joongheon Kim

Korea University, School of Electrical Engineering

https://joongheon.github.io

joongheon@korea.ac.kr

Class Schedule and Grading Criteria

Week	In-Classroom	Lab	ETC
01	Syllabus	-	
02	Constants, Variables, and Data Types (Ch. 3); Operators and Expressions (Ch. 4)	-	
03	Managing Input and Output Operators (Ch. 5); Decision Making and Branching (Ch. 6)	-	
04	Decision Making and Looping (Ch. 7)	Lab-01	
05	Array (Ch. 8), Character Arrays and Strings (Ch. 9)	Lab-02	
06	**Quiz (5%)**	Lab-03	
07	User Defined Functions (Ch. 10)	Lab-04	
08	**MIDTERM EXAM (25%)**		
09	Structures and Unions (Ch. 11)	Lab-05	
10	Introduction to Computing (Ch. 1); Pointers (Ch. 12)	Lab-06	
11	Pointers (Supplementary)	Lab-07	
12	File Management in C (Ch. 13)	Lab-08	
13	Dynamic Memory Allocation and Linked Lists (Ch. 14)	Lab-09	
14	The Preprocessor (Ch. 15); Special Topics (Supplementary)	Lab Exam Exercise	
15	Special Topics (Supplementary)	**LAB EXAM (30%)**	
16	**FINAL EXAM (30%)**		

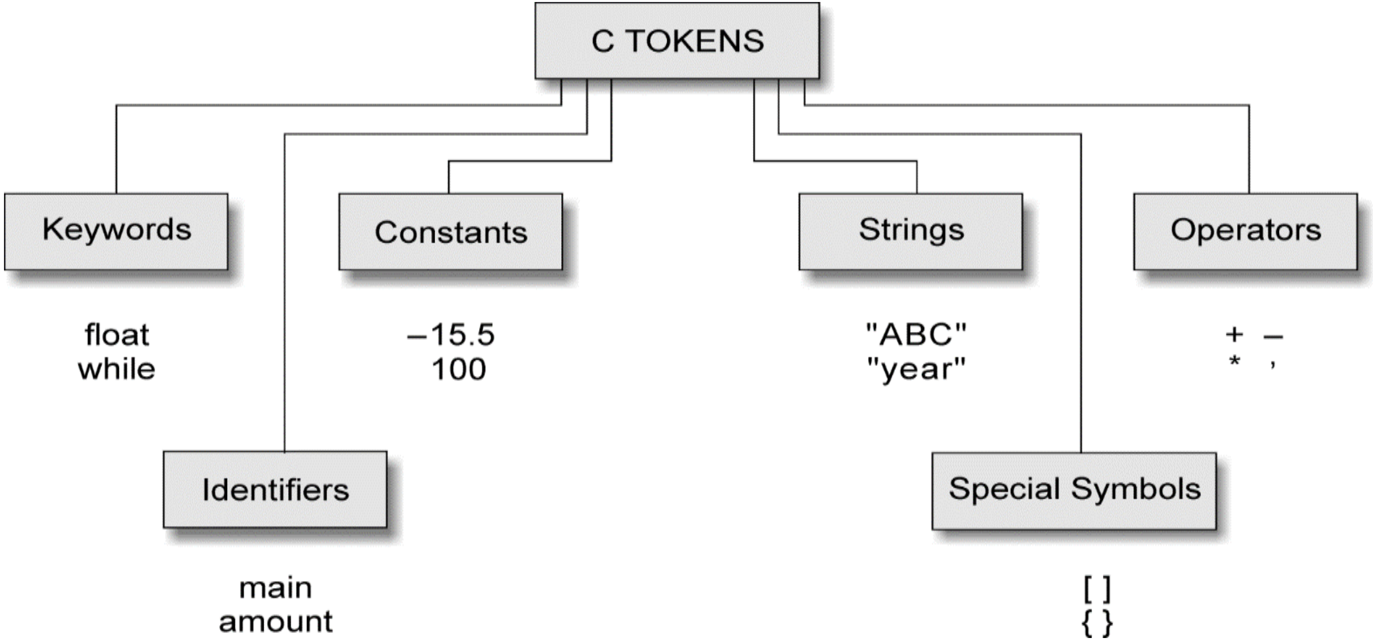
Attendance and Attitude (10%)

Constants, Variables, and Data Types

- Learning Objectives
  - **Know the C character set and keywords**
  - Describe constants and variables
  - Identify the various C data types
  - Discuss how variables are used in a program
  - Explain how constants are used in a program

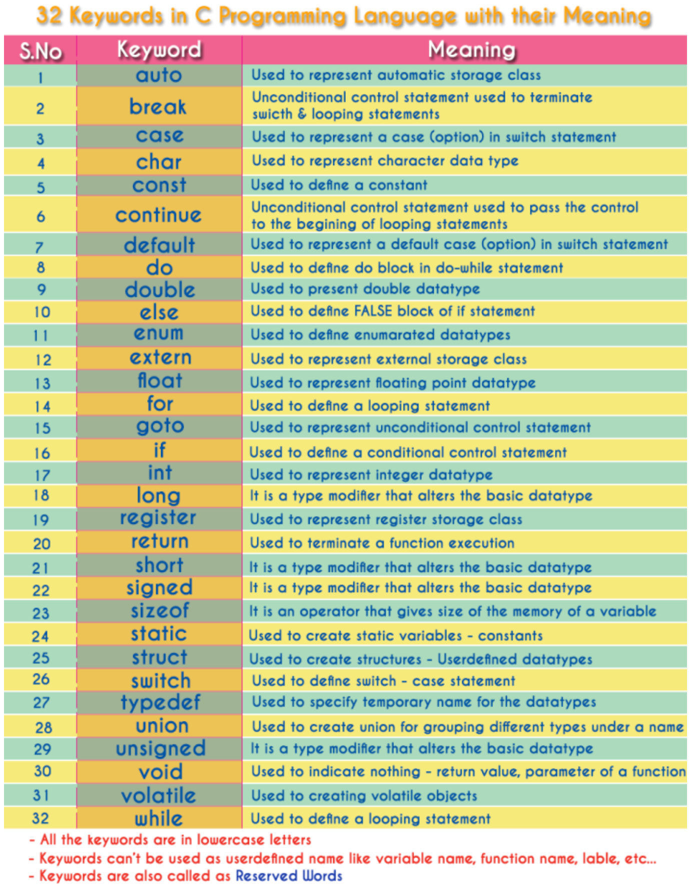
- **Character Set**
  - The characters in C are grouped into the following categories:
    - Letters
    - Digits
    - Special characters
    - White spaces

- **C Tokens**


C TOKENS
Keywords
float
while
Identifiers
main
amount
Constants
–15.5
100
Strings
"ABC"
"year"
Special Symbols
[ ]
{ }
Operators
+ –
* ,

Constants, Variables, and Data Types

- **Keywords**



32 Keywords in C Programming Language with their Meaning

S.No	Keyword	Meaning
1	auto	Used to represent automatic storage class
2	break	Unconditional control statement used to terminate swicth & looping statements
3	case	Used to represent a case (option) in switch statement
4	char	Used to represent character data type
5	const	Used to define a constant
6	continue	Unconditional control statement used to pass the control to the begining of looping statements
7	default	Used to represent a default case (option) in switch statement
8	do	Used to define do block in do-while statement
9	double	Used to present double datatype
10	else	Used to define FALSE block of if statement
11	enum	Used to define enumarated datatypes
12	extern	Used to represent external storage class
13	float	Used to represent floating point datatype
14	for	Used to define a looping statement
15	goto	Used to represent unconditional control statement
16	if	Used to define a conditional control statement
17	int	Used to represent integer datatype
18	long	It is a type modifier that alters the basic datatype
19	register	Used to represent register storage class
20	return	Used to terminate a function execution
21	short	It is a type modifier that alters the basic datatype
22	signed	It is a type modifier that alters the basic datatype
23	sizeof	It is an operator that gives size of the memory of a variable
24	static	Used to create static variables - constants
25	struct	Used to create structures - Userdefined datatypes
26	switch	Used to define switch - case statement
27	typedef	Used to specify temporary name for the datatypes
28	union	Used to create union for grouping different types under a name
29	unsigned	It is a type modifier that alters the basic datatype
30	void	Used to indicate nothing - return value, parameter of a function
31	volatile	Used to creating volatile objects
32	while	Used to define a looping statement

- All the keywords are in lowercase letters
- Keywords can't be used as userdefined name like variable name, function name, lable, etc...
- Keywords are also called as Reserved Words

Constants, Variables, and Data Types

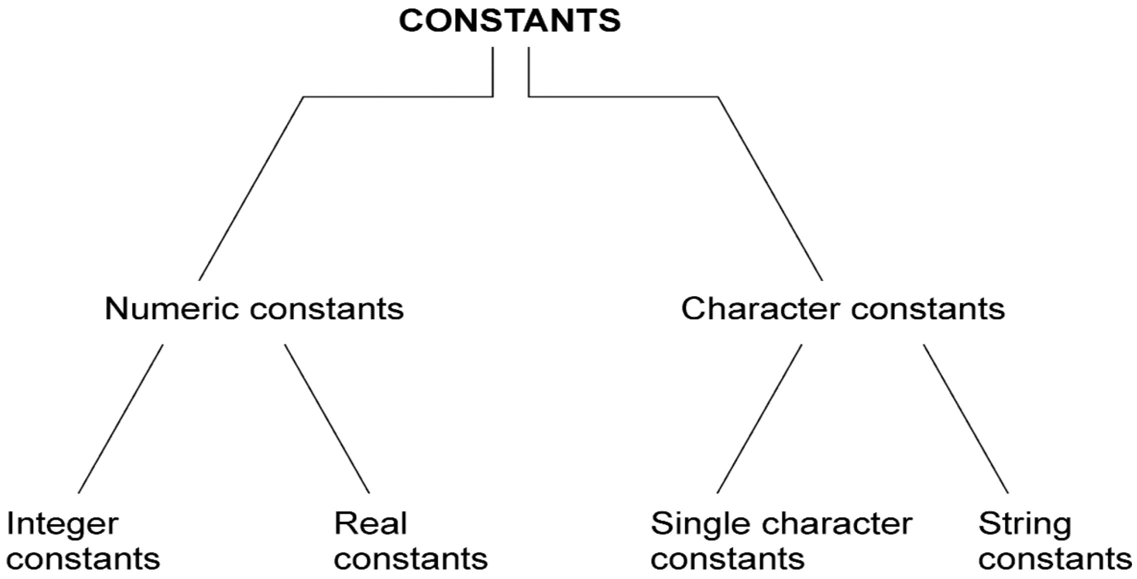
- **Identifiers**
  - Rules
    - First character must be an alphabet (or underscore)
    - Must consist of only letters, digits, or underscore
    - Only first 31 characters are significant
    - Cannot use a keyword
    - Must not contain white space

Constants, Variables, and Data Types

- Learning Objectives
  - Know the C character set and keywords
  - **Describe constants and variables**
  - Identify the various C data types
  - Discuss how variables are used in a program
  - Explain how constants are used in a program

Constants, Variables, and Data Types

- **Constants**
  - Fixed value that do not change during the execution of a program
  - C supports several types of constants

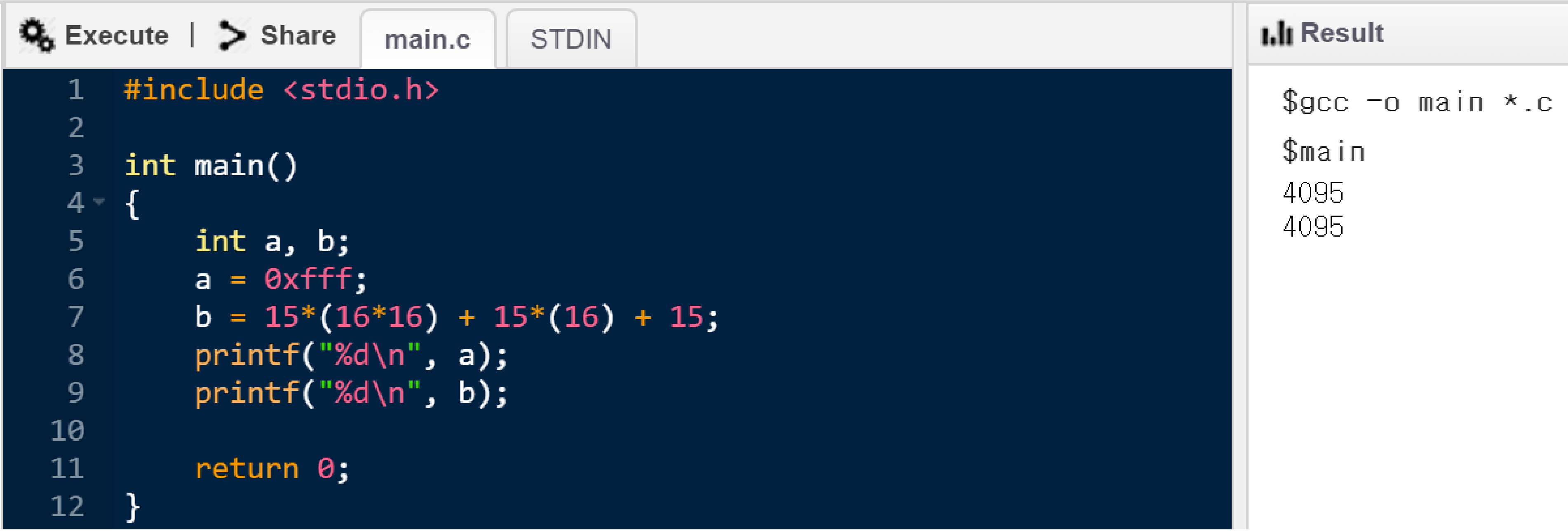


- **Integer Constants**
  - A sequence of digits
  - Three types of integers: decimal, octal, and hexadecimal
    - **Decimal**
      - A set of digits, 0 through 9, preceded by an optional – or + sign
      - Embedded spaces, commas, non-digit characters are not permitted between digits
      - Examples) 123, -321, 0, 654321, +78
    - **Octal**
      - Any combination of digits from the set 0 through 7, with a leading 0
      - Examples) 037, 0, 0435, 0551
    - **Hexadecimal**
      - A sequence of digits preceded by 0x or 0X, may also include alphabets A (or a) through F (or f), where the letter A through F represent the numbers 0 through 15
      - Examples) 0X2, 0x9F, 0Xbcd, 0x

Constants, Variables, and Data Types

Integer Constants

- Examples)
  - **0xFFF = 15*(16*16) + 15*(16) + 15 = 4095**

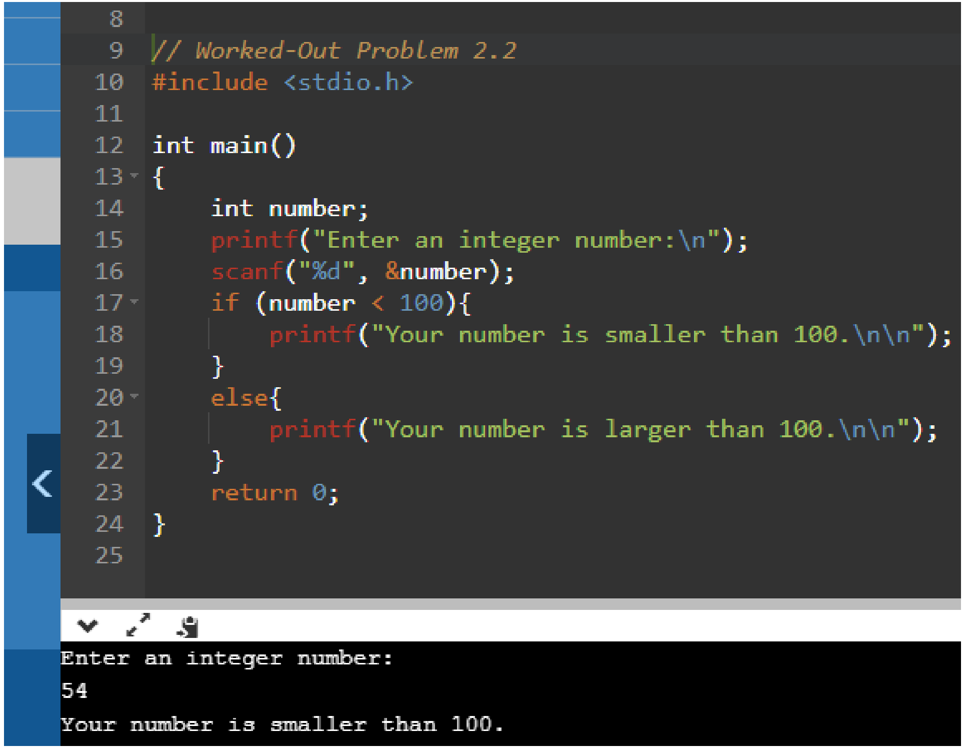


```c
#include <stdio.h>

int main()
{
    int a, b;
    a = 0xfff;
    b = 15*(16*16) + 15*(16) + 15;
    printf("%d\n", a);
    printf("%d\n", b);

    return 0;
}
```

```
$gcc -o main *.c
$main
4095
4095
```

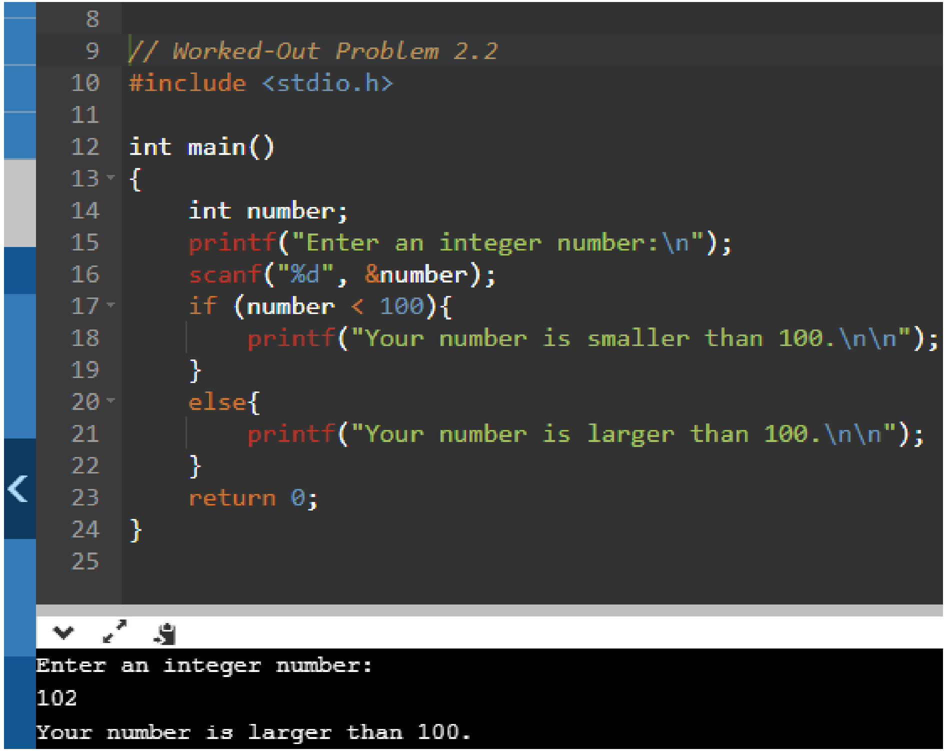
Constants, Variables, and Data Types

- Use other one: https://www.onlinegdb.com/online_c_compiler



```c

// Worked-Out Problem 2.2
#include <stdio.h>

int main()
{
    int number;
    printf("Enter an integer number:\n");
    scanf("%d", &number);
    if (number < 100){
        printf("Your number is smaller than 100.\n\n");
    }
    else{
        printf("Your number is larger than 100.\n\n");
    }
    return 0;
}

```

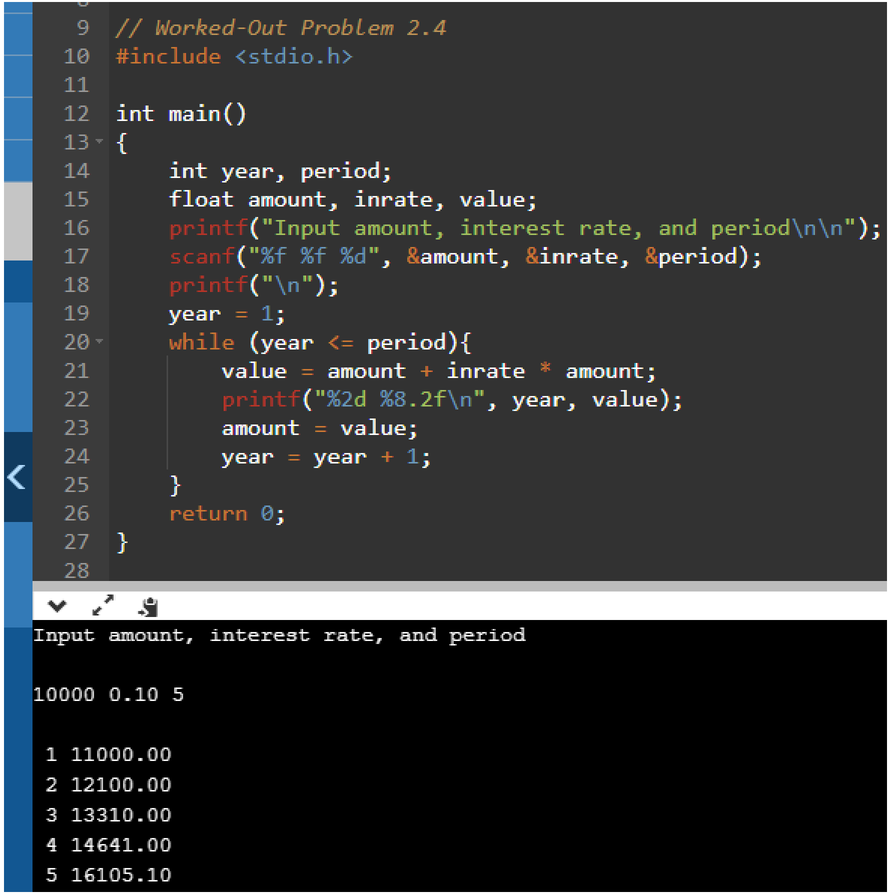
```
Enter an integer number:
54
Your number is smaller than 100.
```



```c

// Worked-Out Problem 2.2
#include <stdio.h>

int main()
{
    int number;
    printf("Enter an integer number:\n");
    scanf("%d", &number);
    if (number < 100){
        printf("Your number is smaller than 100.\n\n");
    }
    else{
        printf("Your number is larger than 100.\n\n");
    }
    return 0;
}

```

```
Enter an integer number:
102
Your number is larger than 100.
```

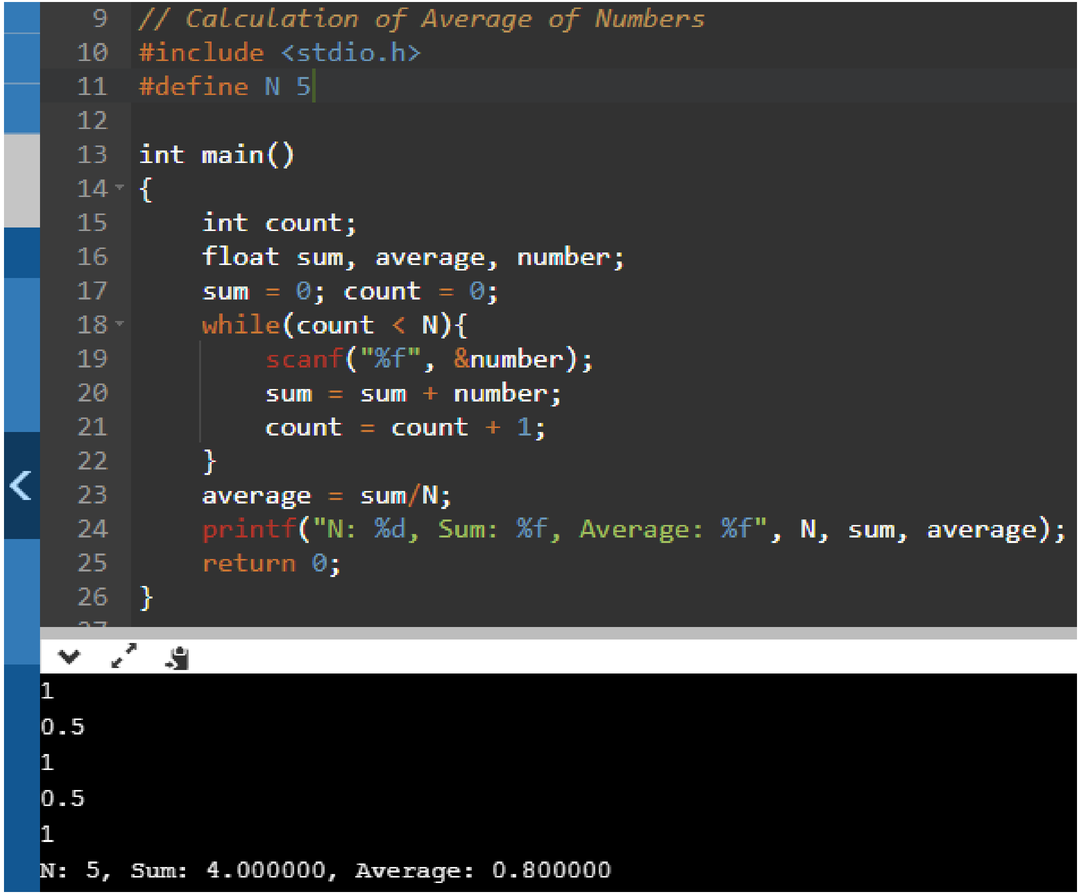
Constants, Variables, and Data Types



```c
// Worked-Out Problem 2.4
#include <stdio.h>

int main()
{
    int year, period;
    float amount, inrate, value;
    printf("Input amount, interest rate, and period\n\n");
    scanf("%f %f %d", &amount, &inrate, &period);
    printf("\n");
    year = 1;
    while (year <= period){
        value = amount + inrate * amount;
        printf("%2d %8.2f\n", year, value);
        amount = value;
        year = year + 1;
    }
    return 0;
}
```

```
Input amount, interest rate, and period

10000 0.10 5

 1 11000.00
 2 12100.00
 3 13310.00
 4 14641.00
 5 16105.10
```

Constants, Variables, and Data Types



```c
// Calculation of Average of Numbers
#include <stdio.h>
#define N 5

int main()
{
    int count;
    float sum, average, number;
    sum = 0; count = 0;
    while(count < N){
        scanf("%f", &number);
        sum = sum + number;
        count = count + 1;
    }
    average = sum/N;
    printf("N: %d, Sum: %f, Average: %f", N, sum, average);
    return 0;
}
```

```
1
0.5
1
0.5
1
N: 5, Sum: 4.000000, Average: 0.800000
```