VG101 Lab Worksheet

Lab 5

Instructor: Dr. Yifei ZHU
TA: Hangrui CAO
TA: Haoxuan SHAN
TA: Qinhang WU
TA: Muchen XU

Code Quality Requirement

- Ensure proper indentation
- Make naming meaningful
- Include necessary comments
- Split the code over functions
- Test the code as much as you can

Basic Syntax Exercises
- 🦖Hello World!
- Output Quotation Mark
- Introduction to ASCII
- Uppercase Converter
- A+B Problem
- Variable Swap
- Triangle
- Days Matter
- BMI Calculator
- Maximal Number
- Fibonacci Number
- Number Pyramid

Tricks in C
- Logical Expression in C
- "Return Value" of Assignment
- Shorthand Operator
- Short Circuit Operator
- Mechanism of FOR
- Increment and Decrement Operators
- Naughty Switch

Appendix
- C Operator Precedence

Basic Syntax Exercises

🦖Hello World!

Write a program to print `Hello World` to the console.

`#printf`

Output Quotation Mark

Try to output double quotation mark with `printf`.

* Use search engine wisely.

`#printf`

Introduction to ASCII

Try this code yourself and check the output.

```
for ( int i = 0 ; i < 255 ; ++ i )
    printf("%c ",char(i));
```

* You may refer to https://theasciicode.com.ar/ to review what ASCII is.

`#char`

Uppercase Converter

Write a program to ask users to input an alphabetic character, and print out the capitalized version of this character.

`#char`

A+B Problem

Write a program, input 2 integers and output their sum.

```
Input:
5 495
Output:
500
```

`#assignment`

Variable Swap

Write a program, input 2 integers, swap and output them respectively.

```
Input:
495 19260817
Output:
19260817 495
```

`#assignment`

Triangle

Input a, b, c, check whether they can be the edge lengths of a triangle. Your output should either be `Yes` or `No`.

`#if-condition`

Days Matter

Input two integers representing a year and a month (later than 1900, earlier than 2020). Find how many days are there in this month.

```
Input:
2020 2
Output:
29
```

`#if-condition`

BMI Calculator

Write a program to calculate the body mass index (BMI) and output the health condition. Input people's weight (in kg) and height (in cm). Notice that the output BMI should be rounded to have precision to 0.1 and the status is determined according to BMI with this precision.
$BMI = \frac{weight_{kg}}{{height_m}^2}$

BMI	Category (Output)
$\leq$ 18.4	"underweight"
18.5 $\leq$ BMI $\leq$ 23.9	"normal"
24.0 $\leq$ BMI $\leq$ 27.9	"Overweight"
$\geq$ 28.0	"Obese"

`#if-condition`

Maximal Number

Given a sequence with n distinct integers, output the maximal number and its index in the sequence.

* Note that you don't need array here.

`#if-condition` `#loop`

```
Input:
10
1 2 3 4 495 5 6 7 8 9
Output:
495 4
```

Fibonacci Number

Output the mth Fibonacci Number. $m < 80$.

You may use the integer type `long long int`.

`#loop` `#assignment`

Number Pyramid

Write a program to output the shape below with an input `n` indicating the layer of the pyramid.

```
Input:
7
Output:
      #
     ###
    #####
   #######
  #########
 ###########
#############
```

`#nested loop`

Tricks in C

Logical Expression in C

Try to assign different logical expressions to integer variables and output them.

```c
int a=1>0;
printf("%d\n", a);
```

Besides, predict the following output.

```c
for (int i=-3;i<=3;i++)
{
    if (i) printf("True: %d", i);
    else printf("False: %d", i);
}
```

"Return Value" of Assignment

Try

```c
int a=0;
int c=(a=10);
printf("%d",c);
```

And explain what will happen if you write

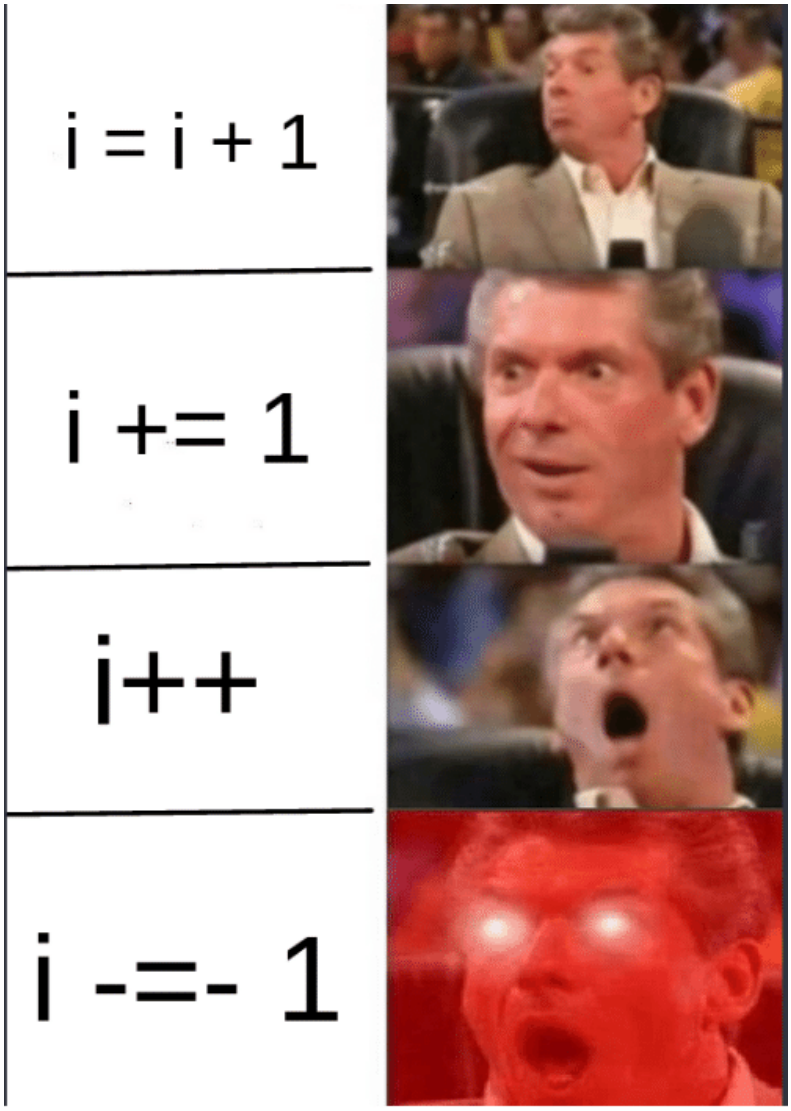
```c
if (a=9) printf("%d",a);
```

And hence conclude that always write like

```c
if (9==a) ...
```

to avoid potential bugs.

Shorthand Operator



How to increment

Short Circuit Operator

```
int a=1,c=0;
// Guess which of these will work
if (c!=0&&a/c==1) printf("%d",a/c);
if (a/c==1&&c!=0) printf("%d",a/c);
```

Mechanism of FOR

Compare the following two program pieces.

```
for(int i=1;i<=10;i++)
    printf("%d",i);
```

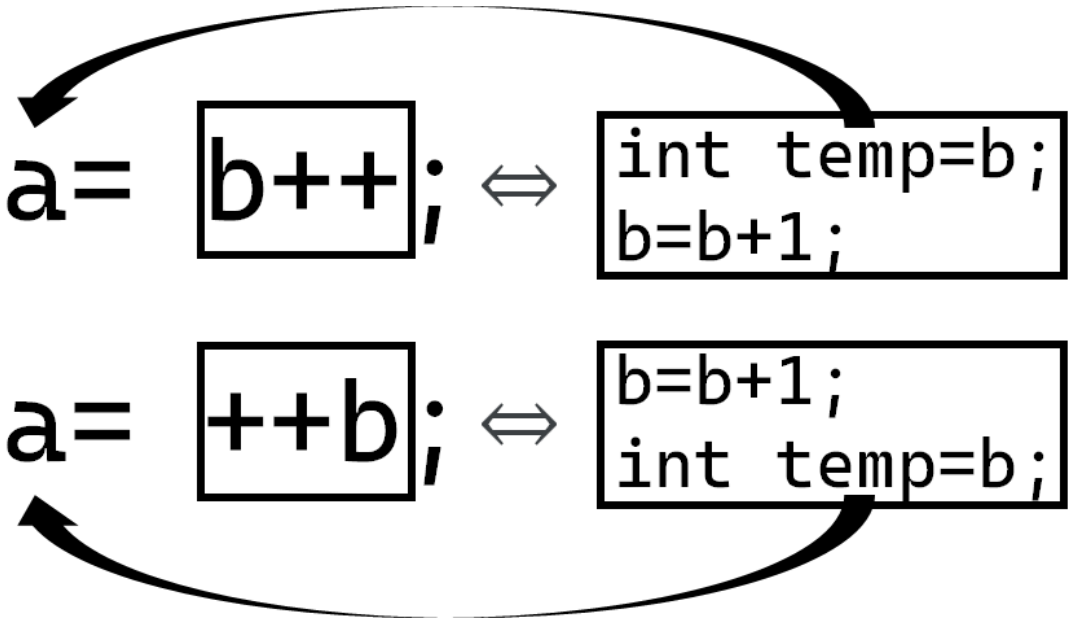
and

```
for(int i=1;i<=10;printf("%d",i))
    i++;
```

And conclude that

```
for(A;B;C)
    D;
/*
During the loop:
A->B->DC->B->DC->...->B->DC->B->Exit
*/
```

Increment and Decrement Operators



And thus predict the output of the following code:

```
for (int i=3;i--;)
    printf("%d ",i);

for (int i=3;--i;)
    printf("%d ",i);
```

Naughty Switch

Copy the following code and record the output. Uncomment four lines of `break` and compare the new output with previous one. What's happening here?

```
int a=1;
switch (a)
{
    case 1:
        a++;
        // break;
    case 2:
        a+=2;
        // break;
```

```
    case 3:
        a=99;
        // break;
    case 4:
        a=-99;
        // break;
}
printf("%d",a);
```

Always remember to check the existence of `break` in a switch statement :-)

Copy the following code and try to fix error(s):

```
int a; scanf("%d",&a);
switch (a)
{
    case 1:
        int b = 1;
        printf("[1] %d", b);
        break;
    case 2:
        b = 2;
        printf("[2] %d", b);
        break;
}
```

Avoid declaring variables inside a switch statement !!!

Appendix

C Operator Precedence

The following table lists the precedence and associativity of C operators. Operators are listed top to bottom, in descending precedence.

Precedence	Operator	Description	Associativity
1	`++` `--`	Suffix/postfix increment/decrement	Left-to-right
	`()`	Function call	
	`[]`	Array subscripting	
	`.`	Structure and union member access	
	`->`	Structure and union member through pointer	
	`(type){list}`	Compound literal(C99)	
2	`++` `--`	Prefix increment and decrement	Right-to-left
	`+` `-`	Unary plus and minus	
	`!` `~`	Logical NOT and bitwise NOT	
	`(type)`	Cast	
	`*`	Indirection (dereference)	
	`&`	Address-of	
	`sizeof`	Size-of	
	`_Alignof`	Alignment requirement(C11)	
3	`*` `/` `%`	Multiplication, division, and remainder	Left-to-right
4	`+` `-`	Addition and subtraction	
5	`<<` `>>`	Bitwise left shift and right shift	
6	`<` `<=`	For relational operators < and ≤ respectively	
	`>` `>=`	For relational operators > and ≥ respectively	
7	`==` `!=`	For relational = and ≠ respectively	
8	`&`	Bitwise AND	
9	`^`	Bitwise XOR (exclusive or)	
10	`|`	Bitwise OR (inclusive or)	
11	`&&`	Logical AND	
12	`||`	Logical OR	
13	`?:`	Ternary conditional	Right-to-Left
14	`=`	Simple assignment	
	`+=` `-=`	Assignment by sum and difference	
	`*=` `/=` `%=`	Assignment by product, quotient, remainder	
	`<<=` `>>=`	Assignment by bitwise left shift and right shift	
	`&=` `^=` `|=`	Assignment by bitwise AND, XOR, and OR	
15	`,`	Comma	Left-to-right

See more details here!