SUSTech Southern University of Science and Technology

CS111，C Programming

Lab / Basic & Condition Statement

黄嘉炜

huangjw3@mail.sustech.edu.cn

深港微电子学院
SCHOOL OF MICROELECTRONICS

Outline

SUSTech Southern University of Science and Technology

- Review
- Operator: More
- Variable：More
- Assignment

Review：Problem #1

SUSTech Southern University of Science and Technology

```
4    int a=0;
5    float x=0,y=0;
6    float result=0;
7    //input three number
8    scanf("%d,%f,%f",&a,&x,&y);
```

如果输入: 1 2 3

a, x, y 能得到需要数值吗?

Review：Problem #2

SUSTech Southern University of Science and Technology

```
5    int a = 0;
6    float x = 0;
7    float y = 0;
8    scanf("%d %f %f , &a, &x, &y");
```

如果输入: 1 2 3

a, x, y 能得到需要数值吗?

Review: Problem #3

SUSTech Southern University of Science and Technology

```
 5      int Z = 0;
 6      float x = 0;
 7      scanf("%d", Z);
 8      float X = 4 - ((3 * (100 - Z)) / 1600);
 9      printf("%.2f\n", X);
10
11      if (Z > 100 || Z < 1) {
12          printf("error");
13      }
```

Review：Problem #4

SUSTech Southern University of Science and Technology

```
11      if ((year % 400) == 0);
12      {
13          puts("yes");
14      }
15      else if((year % 100) == 0);
16      {
17          puts("no");
18      }
19      else ((year % 4 ) == 0);
20      {
21          puts("yes");
22      }
23      else ((year % 4) !== 0);
24      {
25          puts("no");
26      }
27      return 0;
```

Review：Problem #5

SUSTech Southern University of Science and Technology

```
4       float x,y;
5       scanf("%d",&a);
6       scanf("%f",&x);
7       scanf("%f",&y);
8       if (a==1) {
9           float m;
10          m=(x+y)/2*100;
11          m=(int)m/100;
12          printf("%.2f",m);
13          return 1;
14      }
```

#28	✕ Runtime Error	0
#29	✕ Runtime Error	0

OJ 系统要求：return 0;

What are the possible outcomes of my submission? 😟

Please refer to here for a detailed list of possible status.

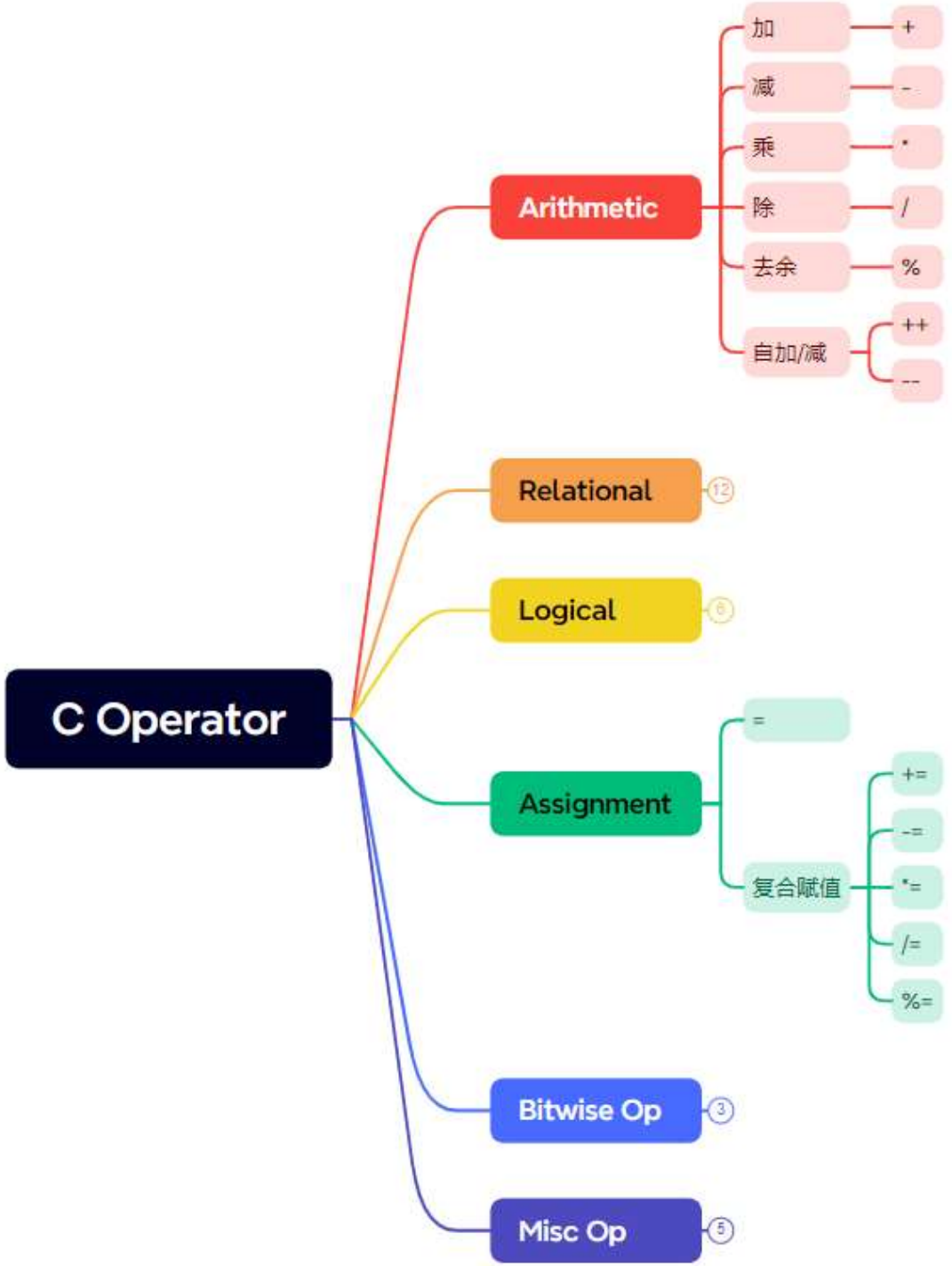
Your program should exit with code `0` to avoid Runtime Error.

Outline

SUSTech Southern University of Science and Technology

- Review
- **Operator: More**
- Variable：More
- Assignment

Operator: More



Operator: showcase，+=，-=，...

```
5    int a = 100;
6    int b = 10;
7
8    a += b;
9    printf("a += b  :=  %d\n", a);
10
11   a -= 10;
12   printf("a -= 10  :=  %d\n", a);
13
14   a *= b;
15   printf("a *= b  :=  %d\n", a);
16
17   a /= 10;
18   printf("a /= 10 :=  %d\n", a);
19
20   a %= b;
21   printf("a %%= b  :=  %d\n", a);
```

复合赋值: add/subtract/multiply/divide and assign

Operator: showcase，+=，-=，...

```
5     int a = 100;
6     int b = 10;
7
8     a += b;
9     printf("a += b  :=  %d\n", a);
10
11    a -= 10;
12    printf("a -= 10  :=  %d\n", a);
13
14    a *= b;
15    printf("a *= b  :=  %d\n", a);
16
17    a /= 10;
18    printf("a /= 10 :=  %d\n", a);
19
20    a %= b;
21    printf("a %%= b  :=  %d\n", a);
```

复合赋值: add/subtract/multiply/divide and assign

等效于：a = a + b;

等效于：a = a - 10;

等效于：a = a * b;

等效于：a = a / 10;

等效于：a = a % b;

```
a += b  :=  110
a -= 10  :=  100
a *= b  :=  1000
a /= 10 :=  100
a %= b  :=  0
```

Operator: showcase，++，--

```
5     int a = 100;
6     int b = 10;
7
8     printf("a++ :=  %d\n", a++);
9     printf("++a :=  %d\n", ++a);
10
11    printf("a-- :=  %d\n", a--);
12    printf("--a :=  %d\n", --a);
```

自加1 / 自减1，注意位置&顺序

- 先加1(或减1)，再取值
- 先取值，后加1(或减1)

Operator: showcase, ++, --

```
5     int a = 100;
6     int b = 10;
7
8     printf("a++ :=  %d\n", a++);
9     printf("++a :=  %d\n", ++a);
10
11    printf("a-- :=  %d\n", a--);
12    printf("--a :=  %d\n", --a);
13
14    int res = a+++--b;
15    printf("a %d, b %d, res %d\n", a, b, res);
```

Precedence	Operator	Description	Associativity
	++ --	Suffix/postfix increment and decrement	Left-to-right
1	()	Function call	
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(*type*){*list*}	Compound literal(C99)	
	++ --	Prefix increment and decrement[note 1]	Right-to-left
2	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(*type*)	Cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of[note 2]	
	_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[note 3]	Right-to-left
	=	Simple assignment	
	+= -=	Assignment by sum and difference	
14[note 4]	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= |=	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

Operator: showcase, ++, --

```c
    int a = 100;
    int b = 10;

    printf("a++ :=  %d\n", a++);
    printf("++a :=  %d\n", ++a);

    printf("a-- :=  %d\n", a--);
    printf("--a :=  %d\n", --a);

    int res = a+++--b;
    printf("a %d, b %d, res %d\n", a, b, res);
```

```
a++ :=  100
++a :=  102
a-- :=  102
--a :=  100
a 101, b 9, res 109
```

Precedence	Operator	Description	Associativity
	++ --	Suffix/postfix increment and decrement	Left-to-right
1	( )	Function call	
	[ ]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(*type*){*list*}	Compound literal(C99)	
	++ --	Prefix increment and decrement[note 1]	Right-to-left
2	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(*type*)	Cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of[note 2]	
	_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[note 3]	Right-to-left
	=	Simple assignment	
14[note 4]	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= |=	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

Outline

SUSTech Southern University of Science and Technology

- Review
- Operator: More
- **Variable：More**
- Assignment

Variable：More integer

SUSTech Southern University of Science and Technology

数据类型	长度	有符号数值范围
long long	8 bytes (64 bits)	-2^63 ~ 2^63-1
long	4 or 8 bytes	... 系统相关
int	4 bytes (32 bits)	-2^31 ~ 2^31-1
short	2 bytes (16 bits)	-2^15 ~ 2^15-1

Appendix, sizeof

Return the size of variable (number of bytes)

```
5     long long l = 2000000000000LL;
6     int i = 2e8;
7     short s = 2e4;
8
9     printf("sizeof(l) %d\n", sizeof(l));
10    printf("sizeof(i) %d\n", sizeof(i));
11    printf("sizeof(s) %d\n", sizeof(s));
12
13    printf("l %lld\n", l);
14    printf("i %d\n", i);
15    printf("s %hd\n", s);
```

注意：输入/输出格式

- **64bit整数对应，%lld**
- **16bit整数对应，%hd**

```
sizeof(l) 8
sizeof(i) 4
sizeof(s) 2
l 2000000000000
i 200000000
s 20000
```

Variable：unsigned

SUSTech Southern University of Science and Technology

数据类型	长度	无符号数值范围
unsigned long long	8 bytes (64 bits)	0 ~ 2^64-1
unsigned int	4 bytes (32 bits)	0 ~ 2^32-1
unsigned short	2 bytes (16 bits)	0 ~ 2^16-1

Variable：unsigned

```
5    // unsigned float f = 1.0;
6    unsigned int ui = 0;
7    unsigned short us = 0;
8
9    scanf("%d %u", &ui, &us);
10   printf("ui %u, %d\n", ui, ui);
11   printf("us %u, %d\n", us, us);
```

注意：浮点数不能用 unsigned

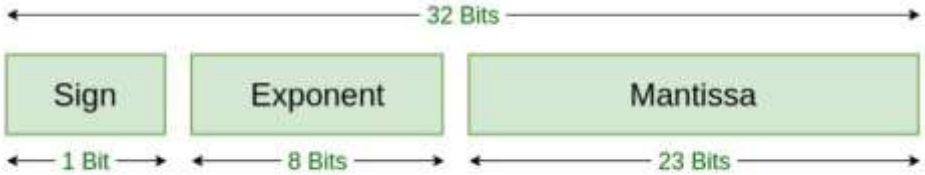
注意：无符号整型的输入/输出格式，%u

```
100 100
ui 0, 0
us 100, 100
```

Variable：More Real

数据类型	长度	数值范围
double	8 bytes (64 bits)	双精度浮点型 1.7×10^{-308} to $1.7\times10^{+308}$
float	4 bytes (32 bits)	单精度浮点型 3.4×10^{-38} to 3.4×10^{38}

Double Precision
IEEE 754 Floating-Point Standard



Single Precision
IEEE 754 Floating-Point Standard

Variable：More Real

数据类型	长度	数值范围
double	8 bytes (64 bits)	双精度浮点型 1.7×10^{-308} to 1.7×10^{308}
float	4 bytes (32 bits)	单精度浮点型 3.4×10^{-38} to 3.4×10^{38}

Can up to **15** decimal points without loss of precision

Can up to **7** ……

```
6     // declaring and initializing
7     float f = 9.12345678f;
8     double d = 9.12345678;
9
10    //%lf or %f both can be used to print Float values
11    printf("float value is %.8f\n", f);
12    printf("double value is %.8lf\n", d);
```

```
float value is 9.12345695
double value is 9.12345678
```

Variable：More Real

数据类型	长度	数值范围
double	8 bytes (64 bits)	双精度浮点型 1.7×10^{-308} to 1.7×10^{308}
float	4 bytes (32 bits)	单精度浮点型 3.4×10^{-38} to 3.4×10^{38}

Can up to **15** decimal points without loss of precision

Can up to **7** ……

```
6       // declaring and initializing
7       float f = 1009.12345678f;
8       double d = 1009.12345678;
9
10      //%lf or %f both can be used to print Float values
11      printf("float value is %.8f\n", f);
12      printf("double value is %.8lf\n", d);
```

```
float value is 1009.12347412
double value is 1009.12345678
```

Variable：char

SUSTech Southern University of Science and Technology

```
5     char c0, c1, c2, c3;
6     c0 = 'a';
7     c1 = 'A';
8     c2 = '1';
9     c3 = '+';
10    printf("sizeof(char) %d\n", sizeof(char));
11    printf("c0 %c, c1 %c, c2 %c, c3 %c\n", c0, c1, c2, c3);
```

Variable：ASCII encoding

SUSTech Southern University of Science and Technology

ASCII (American Standard Code for Information Interchange)

1. **ASCII control characters (0-31 and 127)**
2. **ASCII printable characters (32-126) (most commonly referred to)**
3. **Extended ASCII characters (128-255)**

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	033	00100001	A	065	01000001	a	097	01100001
"	034	00100010	B	066	01000010	b	098	01100010
#	035	00100011	C	067	01000011	c	099	01100011
$	036	00100100	D	068	01000100	d	100	01100100
.....								
7	055	00110111	W	087	01010111	w	119	01110111
8	056	00111000	X	088	01011000	x	120	01111000
9	057	00111001	Y	089	01011001	y	121	01111001
:	058	00111010	Z	090	01011010	z	122	01111010

Ref, https://www.geeksforgeeks.org/ascii-table

Variable：char can as integer

SUSTech Southern University of Science and Technology

```
5     char c0, c1, c2, c3;
6     c0 = 'a';
7     c1 = 'A';
8     c2 = '1';
9     c3 = '+';
10    printf("sizeof(char) %d\n", sizeof(char));
11    printf("c0 %c, c1 %c, c2 %c, c3 %c\n", c0, c1, c2, c3);
12    printf("c0 %d, c1 %d, c2 %d, c3 %d\n", c0, c1, c2, c3);
```

```
c0 a, c1 A, c2 1, c3 +
c0 97, c1 65, c2 49, c3 43
```

Variable：char can as integer

SUSTech Southern University of Science and Technology

```c
char c = 97;
printf("c val %d, character %c\n", c, c);
```

```
c val 97, character a
```

Variable：Type casting / Explicit (显式)

```
6     // step1, input 3 values
7     int a;
8     float x;
9     float y;
10    scanf("%d %f %f", &a, &x, &y);
11
12    // step2, based on a, do caculation
13    float result;
14    if (a == 1) {
15        result = (int)(x + y) / 2;
16    } else if (a == -1) {
17        result = ((int)x + (int)y) / 2;
18    } else {
19        result = (x + y) / 2;
20    }
```

(DataType) (expression)

(DataType) variable

Variable：Type casting / Implicit (隐式)

SUSTech Southern University of Science and Technology

```
5     int a = 1;
6
7     float f = a / 2;
8     printf("a / 2 := %f\n", f);
9
10    f = a / 2.0;
11    printf("a / 2.0f := %f\n", f);
```

Variable：Type casting / Implicit (隐式)

SUSTech Southern University of Science and Technology

```
5     int a = 1;
6
7     float f = a / 2;
8     printf("a / 2 := %f\n", f);
9
10    f = a / 2.0;
11    printf("a / 2.0f := %f\n", f);
```

f = (float) (a / 2);

f = (float)a / 2.0;

```
a / 2 := 0.000000
a / 2.0f := 0.500000
```

Appendix, printf 参数需匹配输出格式

SUSTech Southern University of Science and Technology

```
14      if (a == 1) {
15          // result = (int)(x + y) / 2;
16          printf("%.2f\n", (int)(x + y) / 2);
17      } else if (a == -1) {
18          // result = ((int)x + (int)y) / 2;
19          printf("%.2f\n", ((int)x + (int)y) / 2);
20      } else {
21          // result = (x + y) / 2;
22          printf("%.2f\n", (x + y) / 2);
23      }
```

错误做法！

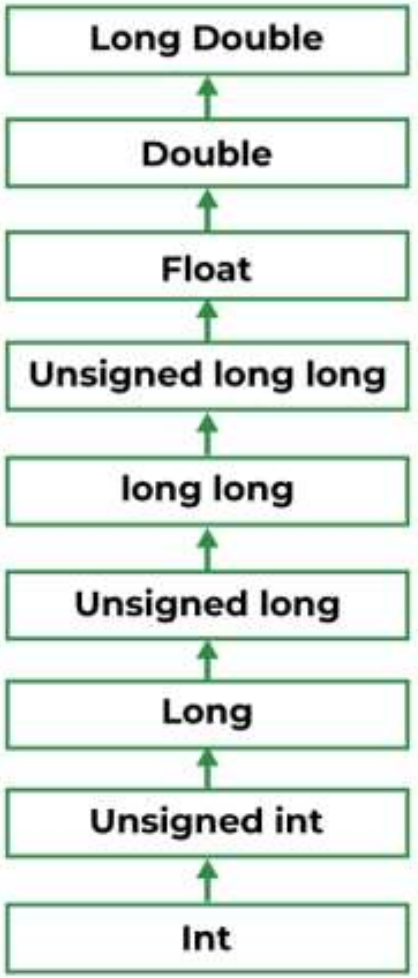
```
1 1 1
0.00
```

Variable：Type casting / 数据丢失问题

SUSTech Southern University of Science and Technology

```
13      int intVal = 2e8; // 2 x 10^8
14      printf("intVal := %d\n", intVal);
15
16      short shortVal = intVal;
17      printf("shortVal := %d\n", shortVal);
```

```
intVal := 200000000
shortVal := -15872
```



Outline

SUSTech Southern University of Science and Technology

- Review
- Operator: More
- Variable： More
- **Assignment**

Assignment 1)

Input three integer variables a, b, and c with respective value ranges from：$-2^{15} \sim 2^{15}-1$, write a program that performs the following computations and outputs:

- **Result 1: Safely calculate and output the integer value of b^2-4ac**
  - Note: ensure no integer overflow.
- **Results 2 and 3: Solve the quadratic equation $ax^2+bx+c=0$ and output the two solutions as double-precision floating-point numbers (保留5位小数).**
  - If the equation has **no real roots**, combine and output the string "error".
  - If the equation has a repeated root (meaning there's only one solution), output that **single solution** only.
  - Note: For square root operations, include the <math.h> header file and utilize the sqrt() function.

```
1 2 3
-8 error
```

```
1 4 3
4 -3.00000 -1.00000
```

如有2个解，先输出小的

```
0 4 3
16 -0.75000
```

Assignment 1)

SUSTech Southern University of Science and Technology

Input three integer variables a, b, and c with respective value ranges from：$-2^{15} \sim 2^{15}-1$, write a program that performs the following computations and outputs:

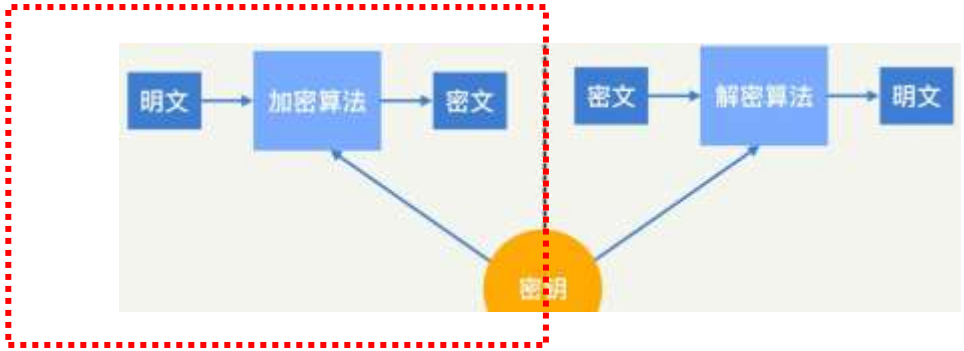
- **Result 1: Safely calculate and output the integer value of b^2-4ac**
  - Note: ensure no integer overflow.
- **Results 2 and 3: Solve the quadratic equation $ax^2+bx+c=0$ and output the two solutions as double-precision floating-point numbers (保留5位小数).**
  - If the equation has **no real roots**, combine and output the string "error".
  - If the equation has a repeated root (meaning there's only one solution), output that **single solution** only.
  - Note: For square root operations, include the <math.h> header file and utilize the sqrt() function.

扩展思考：尝试不使用 math.h 中 sqrt 函数；采用二分法近似求解出平方根，中间语法需要用到循环(e.g. while) 语句。

Assignment 2)

Write a program

- **Input: two 32-bits unsigned integers representing message (msg, ranging from 0 to 999,999) and a key**
- **Output： A four-letter(c3, c2, c1, c0) encrypted password (which only including a-z, A-Z)**




```
000000 1000
iQAm
```

```
666666 10
uVxu
```

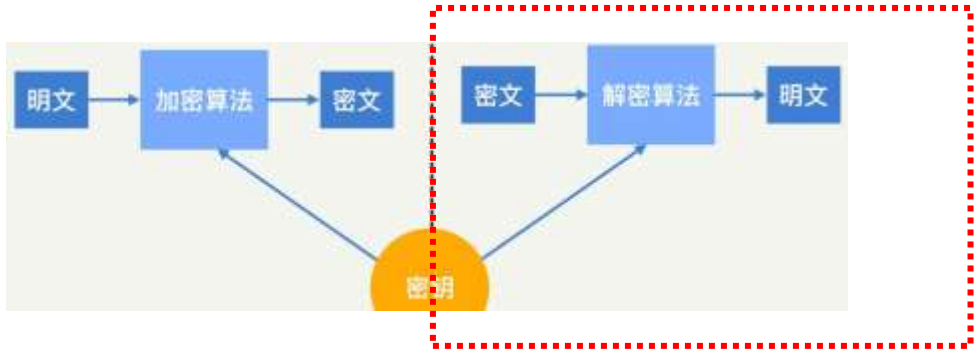
Algorithm details

- **Define four integer index variables i3, i2, i1, and i0, each corresponding to a letter in the password, with values ranging from 0 to 51, where:**
  - 0 through 25 map to lowercase letters a-z
  - 26 through 51 map to uppercase letters A-Z
- **Determine the integer indices for each letter using the following calculations in order:**
  - i0 = ((key++) + (msg % 32)) % 52
  - i1 = (2 * (key++) + (msg / 32 % 32)) % 52
  - i2 = (3 * (key++) + (msg / 1024 % 32)) % 52
  - i3 = (4 * (key++) + (msg / 32768 % 32)) % 52
- **Output the encrypted password with c3 being the first letter, and c0 being the last letter**

Assignment 2)

Write a program

- **Input: two 32-bits unsigned integers representing message (msg, ranging from 0 to 999,999) and a key**
- **Output： A four-letter(c3, c2, c1, c0) encrypted password (which only including a-z, A-Z)**



扩展思考

- **打乱 i0, i1, i2, i3 计算顺序，会影响结果吗?**
- **算法可逆吗?**
- **尝试采用循环(e.g. while / for) 的写法**

Algorithm details

- **Define four integer index variables i3, i2, i1, and i0, each corresponding to a letter in the password, with values ranging from 0 to 51, where:**
  - 0 through 25 map to lowercase letters a-z
  - 26 through 51 map to uppercase letters A-Z
- **Determine the integer indices for each letter using the following calculations in order:**
  - i0 = ((key++) + (msg % 32)) % 52
  - i1 = (2 * (key++) + (msg / 32 % 32)) % 52
  - i2 = (3 * (key++) + (msg / 1024 % 32)) % 52
  - i3 = (4 * (key++) + (msg / 32768 % 32)) % 52
- **Output the encrypted password with c3 being the first letter, and c0 being the last letter**

Appendix, 数学运算函数：开方 / sqrt

C | Numerics | Common mathematical functions

sqrt, sqrtf, sqrtl

Defined in header <math.h>

```
float       sqrtf( float arg );
double      sqrt( double arg );
```

Parameters

arg - floating point value

Return value

If no errors occur, square root of arg ($\sqrt{arg}$), is returned.

If a domain error occurs, an implementation-defined value is returned (NaN where supported).

If a range error occurs due to underflow, the correct result (after rounding) is returned.

Error handling

Errors are reported as specified in `math_errhandling`.

Domain error occurs if arg is less than zero.

If the implementation supports IEEE floating-point arithmetic (IEC 60559),

- If the argument is less than -0, `FE_INVALID` is raised and NaN is returned.
- If the argument is +∞ or ±0, it is returned, unmodified.
- If the argument is NaN, NaN is returned

Ref, https://en.cppreference.com/w/c/numeric/math/sqrt

Appendix, 数学运算函数：开方 / sqrt

SUSTech Southern University of Science and Technology

```c
#include <stdio.h>
#include <math.h>

int main()
{
    double val;
    scanf("%lf", &val);

    double result = sqrt(val);
    printf("sqrt(%lf) := %.5lf", val, result);
    return 0;
}
```

```
1234
sqrt(1234.000000) := 35.12834
```

```
-100
sqrt(-100.000000) := -1.#IND0
```

Appendix, 数学运算函数：开方 / sqrt

```c
#include <stdio.h>
#include <math.h>

int main()
{
    float val;
    scanf("%f", &val);

    double result = sqrt(val);
    printf("sqrt(%lf) := %.5lf", val, result);
    return 0;
}
```

C | Numerics | Common mathematical functions

sqrt, sqrtf, sqrtl

Defined in header <math.h>

```
float       sqrtf( float arg );      (1)  (since C99)
double      sqrt( double arg );      (2)
```

思考：编译会报错吗？

SUSTech Southern University of Science and Technology

THANK YOU