

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 int main(void) {
8     int a = 0;
9     int b = 0;
10
11     scanf("%d%d", &a, &b);
12
13     // TODO: calculate the minimum of a and b
14     // code style: space, {, newline, tab vs. spaces
15     // do not ignore { } for single-line statements
16     // google format, format on save
17     // ?:
18     int min = 0;
19     if (a >= b) {
20         min = b;
21     } else {
22         min = a;
23     }
24
25     // conditional operator, ternary operator
26     // int min = a >= b ? b : a;
27
28     printf("min(%d, %d) = %d\n", a, b, min);
29
30     return 0;
31 }
```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6 int main(void) {
7     int a = 0;
8     int b = 0;
9     int c = 0;
10
11     scanf("%d%d%d", &a, &b, &c);
12
13     // TODO: calculate the minimum of a, b and c
14     int min = 0;
15
16     if (a > b) {
17         if (b > c) {
18             min = c;
19         } else { // b <= c and b < a
20             min = b;
21         }
22     } else { // a <= b
23         if (a < c) {
24             min = a;
25         } else { // c <= a <= b
26             min = c;
27         }
28     }
29
30     printf("min(%d, %d, %d) = %d\n", a, b, c, min);
31
32     return 0;
33 }
```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 int main(void) {
8     int year = 0;
9     scanf("%d", &year);
10
11     // TODO: leap year or not
12     // boolean variable
13     int leap = 0;
14     // TODO (hfwei): arrow code
15     if (year % 4 == 0) {
16         if (year % 100 == 0) {
17             if (year % 400 == 0) {
18                 leap = 1;
19             } else { // can be removed
20                 leap = 0;
21             }
22         } else {
23             leap = 1;
24         }
25     } else { // can be removed; // easier case goes first
26         leap = 0;
27     }
28
29     if (leap == 0) {
30         printf("%d is a common year\n", year);
31     } else {
32         printf("%d is a leap year\n", year);
33     }
34
35     return 0;
36 }
```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 int main(void) {
8     int year = 0;
9     scanf("%d", &year);
10
11     int leap = 0;
12
13     if (year % 4 != 0) {
14         leap = 0;
15     } else {
16         if (year % 100 != 0) {
17             leap = 1;
18         } else {
19             if (year % 400 != 0) {
20                 leap = 0;
21             } else {
22                 leap = 1;
23             }
24         }
25     }
26
27     if (leap == 0) {
28         printf("%d is a common year\n", year);
29     } else {
30         printf("%d is a leap year\n", year);
31     }
32
33     return 0;
34 }
```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 int main(void) {
8     int year = 0;
9     scanf("%d", &year);
10
11     int leap = 0;
12
13     // TODO (hfwei): repeated branch body in conditional chain
14     if (year % 4 != 0) {
15         leap = 0;
16     } else if (year % 100 != 0) {
17         leap = 1; // year % 4 == 0 and year % 100 != 0
18     } else if (year % 400 != 0) {
19         leap = 0;
20     } else {
21         leap = 1; // (year % 4 == 0 and year % 100 == 0 and) year % 400
           == 0
22     }
23
24     if (leap == 0) {
25         printf("%d is a common year\n", year);
26     } else {
27         printf("%d is a leap year\n", year);
28     }
29
30     return 0;
31 }
```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6 int main(void) {
7     int year = 0;
8     scanf("%d", &year);
9
10    int leap = 0;
11
12    // TODO: C operator precedence
13    // URL: https://en.cppreference.com/w/c/language/operator\_precedence
14
15    // TODO (hfwei): order of evaluation
16
17    // TODO: short-circuit evaluation
18    // test: year = 25
19    // test: year = 80
20    // test: year = 100
21    // test: year = 400
22    // TODO: ! (year % 100 == 0)
23    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
24        leap = 1;
25    } else {
26        leap = 0;
27    }
28
29    // int leap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0
30    );
31
32    // TODO: leap = 0
33    if (leap == 0) {
34        printf("%d is a common year\n", year);
35    } else {
36        printf("%d is a leap year\n", year);
37    }
38    return 0;
39 }
```

```

1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 #define NUM 5
8
9 int main(void) {
10     // TODO (hfwei): Variable length array (VLA) folded to constant
    array as an extension
11     // const int NUM = 5;
12
13     // index starting from 0
14     // 5 elements (no '\0')
15     // int[] numbers = {23, 56, 19, 11, 78};
16     // variable-sized object may not be initialized
17     // designator: from C99
18     int numbers[NUM] = {23, 56, 19, 11, 78};
19
20     // []: array subscripting operator
21     int min = numbers[0];
22
23     // syntax + semantics
24     // syntax: for (init-clause; condition-expression; iteration-
    expression) loop-statement
25     // semantics: debug!!!
26     // (1): []
27     // (2): i < NUM: not i <= NUM (accessing out-of-bounds; 越界)
28     // (3): int i = 1; since C99 (declaration in for-loop); code in
    standard C library
29     for (int i = 1;
30         i < NUM;
31         i++) {
32         if (numbers[i] < min) {
33             min = numbers[i];
34         }
35     }
36
37     printf("min = %d\n", min);
38
39     return 0;
40 }

```

```
1 //
2 // Created by hfwei on 2023/10/6.
3 //
4
5 #include <stdio.h>
6
7 #define NUM 5
8
9 int main(void) {
10     int numbers[NUM] = { 0 };
11
12     // int i = 0;
13     // &:amp; address-of operator (numbers[i] is an lvalue)
14     for (int i = 0; i < NUM; i++) {
15         scanf("%d", &numbers[i]);
16     }
17
18     // []: array subscripting operator
19     int min = numbers[0];
20
21     // syntax + semantics
22     // syntax: for (init-clause; condition-expression; iteration-
expression) loop-statement
23     // semantics: debug!!!
24     // (1): []
25     // (2): i < NUM: not i <= NUM (accessing out-of-bounds; 越界)
26     // (3): int i = 1; since C99 (declaration in for-loop); code in
standard C library
27     for (int i = 1;
28         i < NUM;
29         i++) {
30         if (numbers[i] < min) {
31             min = numbers[i];
32         }
33     }
34
35     printf("min = %d\n", min);
36
37     return 0;
38 }
```



```
1 # `2-if-for-array`
2
3 ## Additional
4
5 - `Settings` => `Code Style` (Google)
6 - `Settings` => `Action on Save` (Formatting Code)
7 - TODO (hfwei): CLion code template
8
9 ## `min-of-two.c`
10
11 - `if-else`
12 - code style
13   - tab vs. space video
14   - google format
15   - format on save
16 - `?:`: conditional operator; ternary operator
17 - `fmin, fmax` for doubles
18
19 ## `min-of-three.c`
20
21 - nested `if-else`
22 - `if-else` template
23 - comment for `else`
24
25 ## `leap-if-else.c`
26
27 - flowchart
28 - `leap`: 0/1 integer as a flag
29 - `if-else`
30   - easier cases go first
31 - code style
32   - spaces
33 - `==`: 0 == leap
34 - `if (leap == 0)` vs `if (leap != 0)`
35
36 ## `leap-else-if.c`
37
38 - easier cases go first (Flatten Arrow Code)
39
40 ## `leap-elseif.c`
41
42 - `else if` (Cascading If Statements)
43   - `{ }` removed
44   - `if` and `else` in the same line
45   - `Code` => Format Code (Ctrl + Alt + L)
46 - find the iff condition for leap
47
48 ## `leap.c`
49
50 - `&&`, `||` operator
51   - operator precedence (https://en.cppreference.com/w/c/language/operator\_precedence)
52 - short-circuit
```

```
53 - test: 25, 80, 100, 400
54 - TODO: order of evaluation (https://en.cppreference.com/w/c/
    language/eval\_order)
55 - `i = ++i + i++;`
56 - Code improvements
57 - `if`: without `else`
58 - `int leap = (year % 4 == 0 && year % 100 != 0) || (year % 400 ==
    0);`
59 - `?:` in `printf`
60
61 ## `min-array.c`
62
63 - `array`
64 - `array initializer` (0000)
65 - What if uninitialized? (garbage in, garbage out)
66 - designator (Since C99)
67 - `int n[5] = {[4]=5,[0]=1,2,3,4}; // holds 1,2,3,4,5`
68 ```C
69     int a[MAX] = { // starts initializing a[0] = 1, a[1] = 3, ...
70         1, 3, 5, 7, 9, [MAX-5] = 8, 6, 4, 2, 0
71     };
72 ```
73 - `const int NUM`
74 - `#define NUM 5`
75 - `for`
76 - syntax
77 - `for (init-clause; condition-expression; iteration-expression)
    loop-statement`
78 - semantics (CLion debug!!!)
79 - (1): []
80 - (2): i < NUM: not i <= NUM (accessing out-of-bounds; 0000)
81 - (3): int i = 1; since C99 (declaration in for-loop); code in
    standard C library
82
83 ## `min-array-input.c`
84
85 - `array` initializer
86 - designator
87 - What if uninitialized?
88 - input an array
89 - `&numbers[i]`: lvalue
90 - what if `n (NUM)` is known???
```