

```

1  /**
2   * Bubble Sort Algorithm
3   * See https://en.wikipedia.org/wiki/Bubble\_sort
4   *
5   * Created by hengxin on 10/30/21.
6   */
7
8  #include <stdio.h>
9  #include <time.h>
10 #include <stdlib.h>
11
12 #define LEN 100000
13 //int numbers[LEN] = {6, 5, 3, 1, 8, 7, 2, 4};
14 int numbers[LEN] = { 0 };
15
16 int main() {
17     srand(time(NULL));
18     for (int i = 0; i < LEN; i++) {
19         numbers[i] = rand() % LEN;
20     }
21
22     clock_t start = clock();
23     /**
24      * double "for" version
25      */
26     // int swapped;
27     // for (int i = 0; i < LEN; i++) {
28     //     swapped = 0;
29     //     for (int j = 0; j < LEN - 1 - i; j++) {
30     //         if (numbers[j] > numbers[j + 1]) {
31     //             int tmp = numbers[j];
32     //             numbers[j] = numbers[j + 1];
33     //             numbers[j + 1] = tmp;
34     //             swapped = 1;
35     //         }
36     //     }
37     //     if (!swapped) {
38     //         break;
39     //     }
40     // }
41
42     /**
43      * The optimized version runs slower!!!
44      */
45     int swapped;
46     int len = LEN;
47     int new_len;
48     do {
49         swapped = 0;
50         new_len = 0;
51         for (int j = 0; j < len - 1; j++) {
52             if (numbers[j] > numbers[j + 1]) {
53                 int tmp = numbers[j];

```

```
54     numbers[j] = numbers[j + 1];
55     numbers[j + 1] = tmp;
56     swapped = 1;
57     new_len = j + 1;
58 }
59 }
60 // len--;
61 len = new_len;
62 } while (swapped);
63
64 /**
65  * toupper
66  * see https://stackoverflow.com/a/459704
67  */
68 clock_t diff = clock() - start;
69 long sec = diff / CLOCKS_PER_SEC;
70 printf("Sorting time taken %d seconds.\n", sec);
71
72 for (int i = 0; i < LEN; i++) {
73     printf("%d ", numbers[i]);
74 }
75
76 return 0;
77 }
78
```

```

1  /**
2   * Test whether two arrays A and B have an element in common
3   *
4   * Created by hengxin on 10/30/21.
5   */
6
7  #include <stdio.h>
8  #define LEN_A 5
9  #define LEN_B 5
10
11 int a[LEN_A] = {1, 3, 5, 7, 9};
12 int b[LEN_B] = {2, 4};
13
14 int main() {
15     // int found = 0;
16
17     /**
18      * "break" version
19      */
20     // for (int i = 0; i < LEN_A && (! found); i++) {
21     //     for (int j = 0; j < LEN_B; j++) {
22     //         if (a[i] == b[j]) {
23     //             found = 1;
24     //             break;
25     //         }
26     //     }
27     // }
28     //
29     // if (found) {
30     //     printf("Yes.\n");
31     // } else {
32     //     printf("No.\n");
33     // }
34
35     /**
36      * "goto" version
37      */
38     for (int i = 0; i < LEN_A; i++) {
39         for (int j = 0; j < LEN_B; j++) {
40             if (a[i] == b[j]) {
41                 printf("Yes.\n");
42                 goto found;
43             }
44         }
45     }
46     printf("No.\n");
47
48 found:
49
50     return 0;
51 }
52

```

```
1 /**
2  * file: continue.c
3  *
4  * Created by hengxin on 10/30/21.
5  */
6
7 #define LEN 5
8
9 int numbers[LEN] = {-2, 3, -4, 5, 6};
10
11 int main() {
12     for (int i = 0; i < LEN; i++) {
13         if (numbers[i] < 0) { // skip negative elements
14             continue;
15         }
16
17         // if () // skip ...
18
19         // do positive elements (which may be complicated)
20     }
21     return 0;
22 }
23
```

```
1 /**
2  * Count the number of occurrences of each digit,
3  * of white space characters, and of all other characters.
4  *
5  * Created by hengxin on 10/16/21.
6  */
7
8 /**
9  * Count the number of occurrences of each digit,
10 * of white space characters, and of all other characters.
11 */
12
13 #include <stdio.h>
14 #include <ctype.h>
15 #define LEN 10
16
17 int main() {
18     int digit_count[LEN] = {0};
19     int ws_count = 0;
20     int other_count = 0;
21
22     /**
23      * "if" version
24      * Note: fails to run this program in "Run"
25      * See: https://youtrack.jetbrains.com/issue/CPP-5704
26      * Use "Terminal" instead.
27      * Or use the "input redirection" technique
28      */
29     char ch;
30     while (scanf("%c", &ch) != EOF) {
31         if (isdigit(ch)) {
32             digit_count[ch - '0']++;
33         } else if (isspace(ch)) {
34             ws_count++;
35         } else {
36             other_count++;
37         }
38     }
39
40     printf("digit_count:");
41     for (int i = 0; i < LEN; i++) {
42         printf("%d : %d\n", i, digit_count[i]);
43     }
44     printf("\nws_count: %d\n", ws_count);
45     printf("other_count: %d\n", other_count);
46
47     return 0;
48 }
```

```
1  /**
2  * Count the number of occurrences of each digit,
3  * of white space characters, and of all other characters.
4  *
5  * Created by hengxin on 10/16/21.
6  */
7
8  #include <stdio.h>
9  #define LEN 10
10
11 int main() {
12     int digit_count[LEN] = {0};
13     int ws_count = 0;
14     int other_count = 0;
15
16     /**
17      * "switch-case" version
18      * Note: fails to run this program in "Run"
19      * See: https://youtrack.jetbrains.com/issue/CPP-5704
20      * Use "Terminal" instead.
21      * Or use the "input redirection" technique.
22      */
23     char ch;
24     while (scanf("%c", &ch) != EOF) {
25         switch (ch) {
26             case '0': case '1': case '2': case '3': case '4':
27             case '5': case '6': case '7': case '8': case '9':
28                 digit_count[ch - '0']++;
29                 break;
30             case ' ': case '\n': case '\t':
31                 ws_count++;
32                 break;
33             default:
34                 other_count++;
35                 break;
36         }
37     }
38
39     printf("digit_count:");
40     for (int i = 0; i < LEN; i++) {
41         printf(" %d", digit_count[i]);
42     }
43     printf("\nws_count: %d\n", ws_count);
44     printf("other_count: %d\n", other_count);
45
46     return 0;
47 }
48
49
```

```

1  /**
2   * file: game-of-life.c
3   *
4   * Simulate "Conway's Game of Life"
5   * See https://en.wikipedia.org/wiki/Conway%27s\_Game\_of\_Life
6   * Play with it: https://playgameoflife.com/
7   *
8   * Created by hengxin on 10/30/21.
9   */
10
11 #include <stdio.h>
12 #include <unistd.h>
13 #define SIZE 6
14 int board[SIZE][SIZE] = {
15     {0},
16     {0, 1, 1, 0, 0, 0},
17     {0, 1, 1, 0, 0, 0},
18     {0, 0, 0, 1, 1, 0},
19     {0, 0, 0, 1, 1, 0},
20     {0}};
21
22 int main() {
23     int old_board[SIZE + 2][SIZE + 2];
24     for (int row = 0; row < SIZE + 2; row++) {
25         for (int col = 0; col < SIZE + 2; col++) {
26             if (row == 0 || row == SIZE + 1 || col == 0 || col == SIZE + 1
27 ) {
28                 old_board[row][col] = 0;
29             } else {
30                 old_board[row][col] = board[row - 1][col - 1];
31             }
32         }
33     }
34     for (int row = 1; row <= SIZE + 1; row++) {
35         for (int col = 1; col <= SIZE + 1; col++) {
36             printf("%c ", old_board[row][col] ? '*' : ' ');
37         }
38         printf("\n");
39     }
40     printf("\033c");
41
42     int new_board[SIZE + 2][SIZE + 2];
43
44     for (int tick = 1; tick < 10; tick++) {
45         for (int row = 1; row <= SIZE; row++) {
46             for (int col = 1; col <= SIZE; col++) {
47                 int neighbours =
48                     old_board[row - 1][col - 1] +
49                     old_board[row - 1][col] +
50                     old_board[row - 1][col + 1] +
51                     old_board[row][col - 1] +
52                     old_board[row][col + 1] +

```

```

53         old_board[row + 1][col - 1] +
54         old_board[row + 1][col] +
55         old_board[row + 1][col + 1];
56
57     if (old_board[row][col]) {
58         new_board[row][col] = (neighbours == 2 || neighbours == 3);
59     } else {
60         new_board[row][col] = (neighbours == 3);
61     }
62 }
63 }
64
65 for (int row = 1; row <= SIZE; row++) {
66     for (int col = 1; col <= SIZE; col++) {
67         printf("%c ", new_board[row][col] ? '*' : ' ');
68     }
69     printf("\n");
70 }
71
72 /**
73  * sleep
74  * see https://stackoverflow.com/a/10923084
75  */
76 sleep(2);
77 /**
78  * Clear the console (in Linux)
79  * Warning: This is not robust!!! Just for demonstration.
80  * See https://stackoverflow.com/a/43884673/1833118
81  *
82  * In Windows, use
83  *     #include <conio.h>
84  *     clrscr();
85  */
86 printf("\033c");
87
88 for (int row = 0; row < SIZE + 2; row++) {
89     for (int col = 0; col < SIZE + 2; col++) {
90         old_board[row][col] = new_board[row][col];
91     }
92 }
93
94 }
95
96 return 0;
97 }
98
99

```



```
1  /**
2   * Merge two sorted arrays into one
3   *
4   * Created by hengxin on 10/30/21.
5   */
6
7  #include <stdio.h>
8  #include <math.h>
9
10 #define LEN_L 5
11 #define LEN_R 6
12
13 int L[LEN_L] = {1, 3, 5, 7, 9};
14 int R[LEN_R] = {0, 2, 4, 6, 8, 10};
15
16 int main() {
17     int l = 0;
18     int r = 0;
19
20     while (l < LEN_L && r < LEN_R) {
21         if (L[l] <= R[r]) {
22             printf("%d ", L[l]);
23             l++;
24         } else { // L[l] > R[r]
25             printf("%d ", R[r]);
26             r++;
27         }
28     }
29
30     while (l < LEN_L) {
31         printf("%d ", L[l]);
32         l++;
33     }
34     while (r < LEN_R) {
35         printf("%d ", R[r]);
36         r++;
37     }
38
39     return 0;
40 }
41
```