

```
1 // Created by hfwei on 2022/10/13.
2 //
3
4 #include <stdio.h>
5
6 #define LEN 10
7 int dictionary[LEN] = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55};
8
9 int main() {
10     int key = 0;
11     scanf("%d", &key);
12
13     int index = -1;
14
15     int low = 0;
16     int high = LEN - 1;
17
18     while (low <= high) {
19         // int mid = low + (high - low) / 2
20         int mid = (low + high) / 2;
21         printf("low = %d\t high = %d\t mid = %d\n", low, high, mid);
22
23         if (key < dictionary[mid]) {
24             high = mid - 1;
25         } else if (key > dictionary[mid]) {
26             low = mid + 1;
27         } else {
28             index = mid;
29             break;
30             // search for the leftmost one
31             // high = index - 1;
32         }
33     }
34
35     if (index == -1) {
36         printf("Not found!\n");
37     } else {
38         printf("The index of %d is %d.\n", key, index);
39     }
40
41     return 0;
42 }
```



```
1 //
2 // Created by hengxin on 10/19/22.
3 //
4 // For the usage of clock(),
5 // please refer to https://legacy.cplusplus.com/reference/ctime/clock/.
6 //
7
8 #include <stdlib.h>
9 #include <time.h>
10 #include <stdio.h>
11 #include <stdbool.h>
12
13 #define LEN 1000000
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     // set a clock
23     clock_t start = clock();
24
25     bool swapped = true;
26     for (int i = 0; i < LEN && swapped; i++) {
27         swapped = false;
28         for (int j = 0; j < LEN - 1 - i; j++) {
29             if (numbers[j] > numbers[j + 1]) {
30                 int temp = numbers[j];
31                 numbers[j] = numbers[j + 1];
32                 numbers[j + 1] = temp;
33                 swapped = true;
34             }
35         }
36     }
37
38     // record the end clock
39     clock_t end = clock();
40
41     for (int i = 0; i < LEN; i++) {
42         printf("%d ", numbers[i]);
43     }
44
45     long sec = (end - start) / CLOCKS_PER_SEC;
46     printf("Sorting %d numbers in %ld seconds.\n", LEN, sec);
47
48     return 0;
49 }
```



```

1 //
2 // Created by hengxin on 10/19/22.
3 // Run it with "Terminal"
4 //
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <unistd.h>
9
10 #define SIZE 6
11 const int board[SIZE][SIZE] = {
12     {0},
13     {0, 1, 1, 0, 0, 0},
14     {0, 1, 1, 0, 0, 0},
15     {0, 0, 0, 1, 1, 0},
16     {0, 0, 0, 1, 1, 0},
17     {0}
18 };
19
20 //const int board[SIZE][SIZE] = {
21 //    [1][1] = 1, [1][2] = 1,
22 //    [2][1] = 1, [2][2] = 1,
23 //    [3][3] = 1, [3][4] = 1,
24 //    [4][3] = 1, [4][4] = 1
25 //};
26
27 int main() {
28     int old_board[SIZE + 2][SIZE + 2] = {0};
29
30     for (int row = 0; row < SIZE + 2; row++) {
31         for (int col = 0; col < SIZE + 2; col++) {
32             if (row == 0 || row == SIZE + 1 || col == 0 || col == SIZE + 1
33 ) {
34                 old_board[row][col] = 0;
35             } else {
36                 old_board[row][col] = board[row - 1][col - 1];
37             }
38         }
39     }
40     // print the original board
41     for (int row = 1; row <= SIZE; row++) {
42         for (int col = 1; col <= SIZE; col++) {
43             printf("%c ", old_board[row][col] ? '*' : ' ');
44         }
45         printf("\n");
46     }
47
48     // clear the screen
49     system("clear");
50
51     int new_board[SIZE + 2][SIZE + 2] = {0};
52

```

```

53  for (int round = 1; round < 10; round++) {
54      for (int row = 1; row <= SIZE; row++) {
55          for (int col = 1; col <= SIZE; col++) {
56              // count the number of neighbours of old_board[row][col]
57              int neighbours =
58                  old_board[row - 1][col - 1] +
59                  old_board[row - 1][col] +
60                  old_board[row - 1][col + 1] +
61                  old_board[row][col - 1] +
62                  old_board[row][col + 1] +
63                  old_board[row + 1][col - 1] +
64                  old_board[row + 1][col] +
65                  old_board[row + 1][col + 1];
66
67              // evaluate the new board
68              if (old_board[row][col]) { // old_board[row][col] is alive
69                  new_board[row][col] = (neighbours == 2 || neighbours == 3);
70              } else { // old_board[row][col] is dead
71                  new_board[row][col] = (neighbours == 3);
72              }
73          }
74      }
75
76      // print the new board
77      for (int row = 1; row <= SIZE; row++) {
78          for (int col = 1; col <= SIZE; col++) {
79              printf("%c ", new_board[row][col] ? '*' : ' ');
80          }
81          printf("\n");
82      }
83
84      // sleep for a while
85      // Linux: #include <unistd.h>
86      // Windows: #include <windows.h>: Sleep(ms)
87      sleep(1);
88
89      // clear the screen
90      // Windows: #include <conio.h> clrscr();
91      system("clear");
92
93      // start the next round
94      for (int row = 1; row <= SIZE; row++) {
95          for (int col = 1; col <= SIZE; col++) {
96              old_board[row][col] = new_board[row][col];
97          }
98      }
99  }
100
101  return 0;
102 }

```

```

1 //
2 // Created by hfwei on 2022/9/21.
3 //
4
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include <time.h>
8
9 int main() {
10     int high = 100;
11     int number_of_tries = 2;
12
13     /**
14      * (1) print the rules of the game to players
15      */
16     printf("=====\n"
17           "Let us play the Guess the Number game.\n"
18           "The computer will generate a random number between 1 and %d.\n"
19           "You have totally %d tries.\n"
20           "=====\n",
21           high, number_of_tries);
22
23     /**
24      * (2) generate a random number between 1 and high as the secret
25      * number
26      */
27     srand(time(NULL));
28     int secret = rand() % high + 1;
29     printf("====Just for Test: secret = %d.====\n", secret);
30
31     /**
32      * (3) ask the player to input a guess
33      */
34     while (number_of_tries > 0) {
35         printf("You still have %d tries.\n"
36               "Please input your guess.\n", number_of_tries);
37
38         /**
39          * (4) obtain the guessed number,
40          * compare it with the secret number,
41          * and inform the player of the result
42          */
43         int guess;
44         scanf("%d", &guess);
45
46         if (guess == secret) {
47             printf("Congrats! You win! \n");
48             break;
49         } else if (guess > secret) {
50             printf("guess > secret.\n");
51         } else {
52             printf("guess < secret.\n");

```

```
52     }
53
54     /**
55      * (5) repeat (3)-(4) until the player wins or loses
56      */
57     number_of_tries--;
58
59     if (number_of_tries == 0) {
60         printf("Sorry. You Lose!");
61     }
62 }
63
64 return 0;
65 }
```



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



```
1 // Created by hfwei on 2022/10/6.
2 //
3
4 #include <stdio.h>
5 int main() {
6     int year = 0;
7     scanf("%d", &year);
8
9     int leap = 0;
10
11     // test: year = 25
12     // test: year = 80
13     // test: year = 100
14     // test: year = 400
15     if ((year % 4 == 0 && year % 100 != 0) ||
16         (year % 400 == 0)) {
17         leap = 1;
18     }
19
20     // if (leap == 0) {
21     //     printf("The year %d is a common year.\n", year);
22     // } else {
23     //     printf("The year %d is a leap year.\n", year);
24     // }
25
26     printf("The year %d is a %s year.\n", year, leap? "leap" : "common"
27 );
28     return 0;
29 }
30
```



```
1 // Created by hfwei on 2022/10/13.
2 //
3
4 #include <stdio.h>
5 #include <string.h>
6 #include <stdbool.h>
7
8 #define LEN 21
9 char string[LEN] = "";
10
11 int main() {
12     // example: nolemon,nomelon
13     scanf("%20s", string);
14
15     // int len = 0;
16     // while (string[len] != '\0') {
17     //     len++;
18     // }
19     int len = strlen(string);
20     printf("The length of \"%s\" is %d.\n", string, len);
21
22     bool is_palindrome = true;
23     for (int i = 0, j = len - 1; i < j; i++, j--) {
24         if (string[i] != string[j]) {
25             is_palindrome = false;
26             break;
27         }
28     }
29
30     printf("\"%s\" is %s a palindrome.\n", string,
31         is_palindrome ? "" : "not");
32
33     return 0;
34 }
```



```
1 // Created by hfwei on 2022/10/13.
2 //
3
4 #include <stdio.h>
5 #include <stdbool.h>
6 #include <math.h>
7 #include <time.h>
8
9 int main() {
10     int max = 0;
11     scanf("%d", &max);
12
13     int count = 0;
14     clock_t start = clock();
15     for (int number = 2; number <= max; number++) {
16         // why 1? why not 0?
17         // int is_prime = 1;
18         // since C99
19         bool is_prime = true;
20
21         for (int i = 2; i * i <= number; i++) {
22             if (number % i == 0) {
23                 // is_prime = 0;
24                 is_prime = false;
25                 break;
26             }
27         }
28
29         if (is_prime) {
30             count++;
31             printf("%d ", number);
32         }
33     }
34     clock_t end = clock();
35
36     printf("\ncount = %d in %f seconds\n",
37         count, (double) (end - start) / CLOCKS_PER_SEC);
38
39     return 0;
40 }
```



```
1 // Created by hfwei on 2022/10/13.
2 //
3
4 #include <stdio.h>
5
6 #define LEN 20
7 int numbers[LEN] = {0};
8
9 int main() {
10     /*
11      * Input the array
12      * Note: fails to run this program in "Run" (Ctrl + D)
13      * See: https://youtrack.jetbrains.com/issue/CPP-5704
14      * Use "Terminal" instead.
15      */
16     int len = -1;
17     while (scanf("%d", &numbers[++len]) != EOF);
18     // while (scanf("%d", &numbers[++len]) == 1);
19
20     for (int i = 0; i < len; i++) {
21         printf("%d ", numbers[i]);
22     }
23     printf("\n");
24
25     printf("-----\n");
26     for (int i = 0; i < len; i++) {
27         // find the minimum of numbers[i .. len - 1]
28         int min = numbers[i];
29         int min_index = i;
30         for (int j = i + 1; j < len; j++) {
31             if (numbers[j] < min) {
32                 min = numbers[j];
33                 min_index = j;
34             }
35         }
36
37         // swap numbers[i] and numbers[min_index]
38         int tmp = numbers[i];
39         numbers[i] = numbers[min_index];
40         numbers[min_index] = tmp;
41
42         // print it out
43         for (int i = 0; i < len; i++) {
44             printf("%d ", numbers[i]);
45         }
46         printf("\n");
47         printf("-----\n");
48     }
49 }
```



```
1 // file: stars.c
2 // author: hfwei
3 // date: 2022/10/13
4 // description: print stars pyramid
5
6 #include <stdio.h>
7 int main() {
8     int lines;
9     scanf("%d", &lines);
10
11     for (int i = 0; i < lines; ++i) {
12         // print [lines - (i + 1)] spaces
13         for (int j = 0; j < lines - (i + 1); ++j) {
14             printf(" ");
15         }
16
17         // print [2 * i + 1] spaces
18         for (int j = 0; j < 2 * i + 1; ++j) {
19             printf("*");
20         }
21
22         if (i < lines - 1) {
23             printf("\n");
24         }
25     }
26
27     return 0;
28 }
```



```
1 # 5-function
2
3 ## `static.c`
4     - `random.c`
5
6 ## `generic.c`
```