	1	1958	1961	1969	1954	1969

```
File - D:\cpl\2024-cpl-coding\3-for-a-while\stars.c
 1 // Created by hfwei on 2024/10/10.
 3 #include <stdio.h>
 4
 5 int main(void) {
     int lines = 0;
     scanf("%d", &lines);
 7
 8
 9
     // TODO: print stars pyramid
10
     for (int i = 0; i < lines; ++i) {
        // print lines - 1 - i spaces
11
12
        for (int j = 0; j < lines - 1 - i; ++j) {
13
          printf(" ");
        }
14
15
16
        // print 2 * i + 1 stars
17
        for (int j = 0; j < 2 * i + i; ++j) {
         printf("*");
18
        }
19
20
        if (i < lines - 1) {</pre>
21
        printf("\n");
22
23
        }
      }
24
25
26
     return 0;
27 }
```

```
1 // Created by hfwei on 2024/10/10.
2
3 #include <stdio.h>
4 #include <stdbool.h>
5
6 int main(void) {
     int max = 0;
     scanf("%d", &max);
8
9
10
    // TODO: print primes between 1 and max
11
12
     int count = 0;
13
14
     for (int number = 2; number <= max; number++) {</pre>
15
       // decide whether number is α prime
16
       // Since C99: bool (macro extended to _Bool; with
  macros true and false)
17
       // Since C23: will become keywords (bool, true, false
   ); do not need stdbool.h
18
       bool is_prime = true;
       for (int factor = 2; factor * factor <= number; factor</pre>
19
   ++) {
20
         if (number % factor == 0) {
21
           is_prime = false;
22
           break; // test: number = 18
23
         }
24
       }
25
26
       if (is_prime) { // TODO: is_prime == 1; is_prime != 0
27
         count++;
         printf("%d ", number); // TODO: comment this for
28
  performance
29
       }
     }
30
31
     printf("\ncount = %d\n", count);
32
33
34
     return 0;
35 }
```

```
1 # `3-for-a-while
3 ## `stars.c`
4
5 - double loops
6 - for (int i = 0) + for (int j = 0)
8 ## `primes.c`
9
10 - double loops
11 - `int is_prime = 1; `: why 1? why not 0?
12 - `if (is_prime)` vs. `if (is_prime != 0)` vs. `if (
   is_prime == 1)`
13 - testing
- https://www.wolframalpha.com/input?i=+primes+less+than
  +100000
15 - mma: `PrimePi[100000]`
16 - `number = 2`
17 - `break`
18 - `i * i <= number` vs. `i * i < number`</pre>
19 - `stdbool.h`
20 - C89, C99, C23
21 - `bool b = 5`
22 - `(bool) 3.5`
23 - [x] timing
24 - `clock_t start = clock(); clock_t end = clock(); (end
  - start) / CLOCKS_PER_SEC`
25
26 # `binary-search.c`
27
28 - already sorted array
29 - Fib
30 - int index = -1;
31
    - `printf`
32 - `break`
33 - testing
34 - `1`: the leftmost/rightmost one
35 - search for the leftmost/rightmost one
36 - [ ] learn from the standard library???
37 - (low + high) / 2
      - `low + (high - low) / 2`
38
39
      - [ ] try it???
40
41 ## `digits.c`
```

```
42
43 - testing
44 - `do-while`
45
46 ## `selection-sort.c`
47
48 - preparation: scanf
49 - with comments
50 - `swap`
51 - `while (scanf ...)`
    - <a href="https://en.cppreference.com/w/c/io/fscanf">https://en.cppreference.com/w/c/io/fscanf</a>
53
       - Number of receiving arguments successfully assigned
    (which may be zero in case a matching failure occurred
   before
         the first receiving argument was assigned)
54
55
       - or `EOF` if input failure occurs before the first
   receiving argument was assigned
    - How to run this?
56
       - Linux: `Ctrl + D` at the beginning of a line
57
58
      - Mac: `Cmd + D` at the beginning of a line
       - Windows: `Ctrl + Z` at the beginning of a line
59
60 - more `printf` (after each iteration)
61 - `sizeof`
62 - Input&Output indirection
63 - Linux/Windows Cmd
64
65 ## `palindrome.c`
67 - `#define`: pre-processing
68 - `scanf("%20s", string);`
69 - `strlen`
70 - comma expression
71 - `for` version
72 - `while` version
```

```
File - D:\cpl\2024-cpl-coding\3-for-a-while\digits-for.c
 1 // Created by hfwei on 2024/10/10.
 3 #include <stdio.h>
 4
 5 int main(void) {
     int number = 0;
     scanf("%d", &number);
 8
     // Initialize the number of digits to 1
 9
     int num_of_digits = 1;
10
11
     // For numbers other than 0, adjust number of digits
12
     for (; number / 10 != 0; num_of_digits++) {
13
14
        number /= 10;
     }
15
16
17
     printf("Number of digits is %d\n", num_of_digits);
18
19
     return 0;
20 }
```

```
1 // Created by hfwei on 2024/10/10.
2
3 #include <stdio.h>
4 #include <string.h>
5 #include <stdbool.h>
6
7 #define LEN 21
8 char string[LEN] = "";
9
10 int main() {
11 // example: nolemon, nomelon
12
    printf("Input a string containing at most 20 characters.
  \n");
13
   scanf("%20s", string);
14
15 // int len = 0;
16 // while (string[len] != '\0') {
17 // len++;
18 // }
19 int len = strlen(string);
    printf("The length of \"%s\" is %d.\n", string, len);
20
21
22
    // TODO: test palindrome
23
24
    // TODO: the for version
25
    // bool is_palindrome = true;
    // for (int i = 0, j = len - 1; i < j; i++, j--) {
26
    // if (string[i] != string[j]) {
27
         is_palindrome = false;
28
    //
29
    // break;
    // }
30
    // }
31
32
33
    // TODO: the while version
34
    bool is_palindrome = true;
35
36
    int i = 0;
37
    int j = len - 1;
38
    while (i < j) {
39
       if (string[i] != string[j]) {
40
        is_palindrome = false;
41
        break;
42
       }
      i++;
43
```

```
j--;
    }
45
46
    printf("\"%s\" is %s a palindrome.\n", string,
47
            is_palindrome ? "" : "not");
48
49
50
     return 0;
51 }
```

```
1 # for
 2 add_executable(stars stars.c)
3 add_executable(primes primes.c)
4
5 # while (do-while)
6 add_executable(binary-search binary-search.c)
7 add_executable(binary-search-for binary-search-for.c)
8
9 add_executable(digits-while digits-while.c)
10 add_executable(digits-do-while digits-do-while.c)
11 add_executable(digits-for digits-for.c)
12
13 # for-α-while
14 add_executable(selection-sort selection-sort.c)
15 add_executable(palindrome palindrome.c)
```

```
File - D:\cpl\2024-cpl-coding\3-for-a-while\digits-while.c
 1 // Created by hfwei on 2024/10/10.
 3 #include <stdio.h>
 4
 5 int main(void) {
    int number = 0;
     scanf("%d", &number);
 7
 8
 9
     // TODO: number of digits
10
     int num_of_digits = 0;
11
     // TODO: >= (infinite loop)
12
13
     if (number == 0) {
        num_of_digits = 1;
14
     } else {
15
16
        while (number > 0) {
17
          number /= 10;
          num_of_digits++;
18
19
        }
     }
20
21
22
     printf("Number of digits is %d\n",
23
             num_of_digits);
24
25
     return 0;
26 }
```

```
1 // Created by hfwei on 2024/10/10.
3 #include <stdio.h>
4
5 #define LEN 10
6 int dictionary[LEN] = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 };
8 int main(void) {
     int key = 0;
     scanf("%d", &key);
10
11
12
     // TODO: binary search: search for key in dictionary[]
13
     int low = 0;
14
     int high = LEN - 1;
15
16
     int index = -1;
17
18
     while (low <= high) {</pre>
19
       int mid = (low + high) / 2;
20
21
       if (key > dictionary[mid]) {
22
         low = mid + 1;
23
       } else if (key < dictionary[mid]) {</pre>
         high = mid - 1;
24
25
       } else { // key == dictionary[mid]
26
         index = mid;
27
         // break; // what if `break` is removed
28
         high = mid - 1; // find the leftmost index of the
  key
29
       }
     }
30
31
32
     if (index == -1) {
33
       printf("Not found!\n");
34
     } else {
35
       printf("The index of %d is %d.\n", key, index);
36
     }
37
38
     return 0;
39 }
```

```
1 // Created by hfwei on 2024/10/10.
3 #include <stdio.h>
4
5 #define LEN 20
6 int numbers[LEN] = { 0 };
8 int main(void) {
9
    /*
10
     * Input the array
11
12
      * Note: fails to run this program in "Run" (Ctrl + D)
13
     * See: https://youtrack.jetbrains.com/issue/CPP-5704
     * Use "Terminal" instead.
14
15
     *
16
      * TODO: CLion; Terminal
17
      * Linux: Ctrl + D (works now; in the new line, or Ctrl
    + D twice)
      * See https://stackoverflow.com/a/21365313/1833118 (
18
   send and clear the buffer)
      * Windows: Ctrl + Z (does not work on my platform)
19
20
      * TODO: Input&Output redirection
21
          See https://stackoverflow.com/a/11788475/1833118
22
     */
     int len = -1;
23
24
     while (scanf("%d", &numbers[++len]) != EOF);
25
26
     // sizeof numbers / sizeof(numbers[0])
     for (int i = 0; i < len; i++) {</pre>
27
       printf("%d ", numbers[i]);
28
29
     }
30
     printf("\n");
31
32
     // TODO: selection sort
     for (int i = 0; i < len; i++) {</pre>
33
34
       // find the minimum value of numbers[i .. n-1]
35
       int min = numbers[i];
36
       int min_index = i;
37
       for (int j = i + 1; j <= len - 1; ++j) {
38
39
         if (numbers[j] < min) {</pre>
40
           min = numbers[j];
41
           min_index = j;
42
         }
```

```
43
44
       // swap numbers[i] and numbers[min_index]
45
       int temp = numbers[i];
46
       numbers[i] = numbers[min_index];
47
       numbers[min_index] = temp;
48
     }
49
50
51
     for (int i = 0; i < len; i++) {</pre>
       printf("%d ", numbers[i]);
52
     }
53
     printf("\n");
54
55
     return 0;
56
57 }
```

```
1 // Created by hfwei on 2024/10/10.
3 #include <stdio.h>
4
5 int main(void) {
   int number = 0;
     scanf("%d", &number);
7
8
    // TODO: number of digits
9
    int num_of_digits = 0;
10
11
    do {
12
      number /= 10;
13
      num_of_digits++;
14
    } while (number > 0);
15
16
17
     printf("Number of digits is %d\n",
            num_of_digits);
18
19
20
     return 0;
21 }
```

```
1 // Created by hfwei on 2024/10/10.
3 #include <stdio.h>
4
5 #define LEN 10
6 int dictionary[LEN] = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 };
8 int main(void) {
     int key = 0;
     scanf("%d", &key);
10
11
12
     int index = -1;
13
14
     for (int low = 0, high = LEN - 1; low <= high; ) {
       int mid = (low + high) / 2;
15
16
17
       if (key > dictionary[mid]) {
18
         low = mid + 1;
19
       } else if (key < dictionary[mid]) {</pre>
20
         high = mid - 1;
21
       } else { // key == dictionary[mid]
22
         index = mid;
23
         high = mid - 1; // find the leftmost index of the
  key
24
       }
25
     }
26
     if (index == -1) {
27
28
       printf("Not found!\n");
29
     } else {
       printf("The index of %d is %d.\n", key, index);
30
31
     }
32
33
     return 0;
34 }
```