```
File - D:\cpl\2024-cpl-coding\1-types-io\mol.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 7 int main(void) {
      const double MOL = 6.02E23;
      const int GRAM_PER_MOL = 32;
10
11
      int mass = 6;
12
      double quantity = mass * 1.0 / GRAM_PER_MOL * MOL;
13
14
     printf("quantity = %.3e\nquantity = %.5g\n",
15
             quantity, quantity);
16
17
18
      return 0;
```

19 }

```
File - D:\cpl\2024-cpl-coding\1-types-io\admin.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <math.h>
 6 #include <stdio.h>
 7 #include <ctype.h>
 8
 9 int main(void) {
     char first_name[] = "Tayu";
10
11
     char last_name[] = "Lo";
12
13
     char gender = 'm';
14
     // char upper_gender = 'm' + 'A' - 'a';
15
     // printf("upper_gender : %c\n", upper_gender);
16
17
     int birth_year = 1954;
18
     int birth_month = 7;
19
     int birth_day = 20;
20
     char weekday[] = "Tuesday";
21
22
     int c_score = 50;
23
     int music_score = 99;
24
     int medicine_score = 78;
25
26
     double mean = (c_score + music_score + medicine_score
   ) / 3.0;
27
     double sd = sqrt((pow(c_score - mean, 2) +
28
          pow(music_score - mean, 2) +
29
          pow(medicine_score - mean, 2)) / 3.0);
30
31
     int rank = 10;
32
33
     printf("%s %s \t %c\n"
             "%.2d-%d-%d \t %.3s.\n"
34
             "%d \t %d \t %d\n"
35
36
             "%.1f \t %.2f \t %d%%\n",
37
             first_name, last_name, toupper(gender),
38
             birth_month, birth_day, birth_year, weekday,
39
             c_score, music_score, medicine_score,
             mean, sd, rank);
40
41
42
     return 0;
43 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\circle.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 6
 7 int main(void) {
 8 // const: constant
     const double PI = 3.14159;
10
11
     int radius = 10;
12
13
     double circumference = 2 * PI * radius;
14
15
     double area = PI * radius * radius;
16
17 /*
     * format is composed of zero or more directives:
18
     * ordinary characters and conversion specifications
   introduced by %
20
     */
     printf("radius = %d\ncircumference = %.2f\narea = %.2f\n
21
22
             radius, circumference, area);
23
24 return 0;
25 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\sphere.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 6 #include <math.h>
 8 int main(void) {
     const double PI = 3.14159;
10
11
     int radius = 100;
12
     double surface_area = 4 * PI * pow(radius, 2);
13
     double volume = 4.0 / 3 * PI * pow(radius, 3);
14
15
16
     // .4: precision
     // 15: minimum width
17
     // -: flag
18
19
     printf("%-15.4f : surface_area\n%-15.4f : volume\n",
20
             surface_area, volume);
21
22
     return 0;
23 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\admin-scanf.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <math.h>
 6 #include <stdio.h>
 7 #include <ctype.h>
 8
 9 int main(void) {
10
     char first_name[10];
11
     char last_name[10];
12
13
     char gender;
14
15
     int birth_year;
16
     int birth_month;
17
     int birth_day;
18
     char weekday[10];
19
20
     int c_score;
21
     int music_score;
22
     int medicine_score;
23
     int rank;
24
25
     /*
26
     * zero or more directives:
27
      * (1) one or more white-space characters ( , \t, \n);
28
      * (2) ordinary characters (neither % nor white-space
   characters)
29
      * (3) conversion specification introduced by %
30
31
     scanf("%9s%9s %c %d-%d-%d %9s %d%d%d %*lf%*lf %d%%",
32
            first_name, last_name, &gender,
33
            &birth_year, &birth_month, &birth_day, weekday,
34
            &c_score, &music_score, &medicine_score,
35
            &rank);
36
37
     double mean = (c_score + music_score + medicine_score
   ) / 3.0;
38
     double sd = sqrt((pow(c_score - mean, 2) +
39
40
          pow(music_score - mean, 2) +
41
          pow(medicine_score - mean, 2)) / 3.0);
42
```

```
43
     printf("%s %s \t %c\n"
44
            "%.2d-%d-%d \t %.3s.\n"
45
            "%d \t %d \t %d\n"
46
            "%.1f \t %.2f \t %d%%\n",
47
            first_name, last_name, toupper(gender),
48
            birth_month, birth_day, birth_year, weekday,
49
50
            c_score, music_score, medicine_score,
            mean, sd, rank);
51
52
53
     return 0;
54
55
     return 0;
56 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\scanf-error.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 7 int main(void) {
    /*
 9
     * (1) double
     */
10
11
     double d;
     scanf("%f", &d);
12
13
     printf("d = %f\n", d);
14
15
    /*
16
     * (2) using the value of uninitialized variable
17
     */
18
     int age;
     printf("Enter your age: ");
19
20
     scanf("%d", &age);
21
     printf("Your age is %d.\n", age);
22
23
    /*
24 * (3) long string
25
     * segmentation fault: buffer overflow: Process finished
    with exit code -1073741819 (0xC0000005)
26
      */
27
     char name[12];
28
     printf("What's your name? ");
     scanf("%s", name);
29
     printf("Hello %s!\n", name);
30
31
32
     return 0;
33 }
```

```
1 add_executable(circle circle.c)
2
3 add_executable(sphere sphere.c)
4 target_link_libraries(sphere m)
 5
6 add_executable(mol mol.c)
8 add_executable(admin admin.c)
9 target_link_libraries(admin m)
10
11 add_executable(admin-scanf admin-scanf.c)
12 target_link_libraries(admin-scanf m)
13
14 add_executable(printf-error printf-error.c)
15 add_executable(scanf-error scanf-error.c)
16
17 add_executable(scanf-c17-ex3 scanf-c17-ex3.c)
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\printf-error.c
```

```
1 //
2 // Created by hfwei on 2024/9/25.
3 //
4
5 #include <stdio.h>
6
7 int main(void) {
     int secret = 42;
     printf("secret = %f\n", secret);
10
     double pi = 3.14159;
11
     printf("pi = %d\n", pi);
12
13
14
     int light_speed = 299792458;
     printf("light_speed = %c\n", light_speed);
15
16
17
     return 0;
18 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\scanf-c17-ex2.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 7 // Example 2 from C17 (Page 234)
 8 int main(void) {
     int i;
     double x; // original: float x;
10
     char name[50];
11
12
13
     // enter "56789 0123 56a72"
     scanf( "%2d%lf%*d %[0123456789]",
14
15
             &i, &x, name);
16
     // i = 56
17
18
     // x = 789.000000
19
     // name = 56
     printf("i = %d\n"
20
21
             "x = %f \n"
22
             "name = %s\n",
23
             i, x, name);
24
25
     return 0;
26 }
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\scanf-c17-ex3.c
 1 //
 2 // Created by hfwei on 2024/9/25.
 3 //
 4
 5 #include <stdio.h>
 6
 7 // Example 3 from C17 (Page 234)
 8 int main(void) {
     int count = -2;
     double quant = -1; // original: float quant;
10
     char units[21] = "";
11
     char item[21] = "";
12
13
14
     // input:
15 // 2 quarts of oil
16
     // -12.8degrees Celsius
17
   // lots of luck
     // 10.0LBS
18
     // of
19
20
     //
           dirt
     // 100ergs of energy
21
     do {
22
23
       count = fscanf(stdin,
24
                       "%lf%20s of %20s",
25
                       &quant, units, item);
26
       fscanf(stdin, "%*[^\n]");
27
       // added
28
       printf("count = %d\n"
               "quant = %f\n"
29
               "units = %s\n"
30
               "item = %s\n",
31
32
               count, quant, units, item);
33
     } while (!feof(stdin) && !ferror(stdin));
34
     // output:
35
36
     // count = 3;
     // quant = 2; strcpy(units, "quarts"); strcpy(item, "oil
37
   ");
38
     // count = 2; // "C" fails to match "o"
39
     // quant = -12.8; strcpy(units, "degrees");
40
41
     // count = 0; // "l" fails to match "%f"
42
43
```

```
File - D:\cpl\2024-cpl-coding\1-types-io\scanf-c17-ex3.c
44 // count = 3;
45 // quant = 10.0; strcpy(units, "LBS"); strcpy(item, "
   dirt");
46
47
     // count = 0; // "100e" fails to match "%f" // 100.
   000000 rgs energy!!! (not conformed to the C Standard)
48
49 // count = EOF;
50
51
     // A bug in gcc: https://sourceware.org/bugzilla/
   show_bug.cgi?id=1765#c1
52
53
     // input & output:
54
     // 2 quarts of oil
55 // quant = 2.000000
56
     // units = quarts
   // item = oil
57
58
59
     // -12.8degrees Celsius
60
     // quant = -12.800000
61
     // units = degrees
62
     // item = oil
63
64
     // lots of luck
65
     // quant = -12.800000
66
     // units = degrees
     // item = oil
67
68
     // 10.0LBS
69
70
     // of
71
     // dirt
     // quant = 10.000000
72
73
     // units = LBS
74
     // item = dirt
75
76
     // 100ergs of energy
77
     // quant = 100.000000
78
     // units = rgs
79
     // item = energy
80
81
     return 0;
82 }
```

1 56789 0123 56a72

File - D:\cpl\2024-cpl-coding\1-types-io\scanf-c17-ex3-input.txt

- 1 2 quarts of oil
- 2 -12.8degrees Celsius
- 3 lots of luck
- 4 10.0LBS
- 5 of
- 6 dirt
- 7 100ergs of energy