

2020 College of Engineering Systems

Introduction to programming B 【exercise-week04】

Student ID (202018021) Name (Fu, Ziyu)

Answer the following questions. You can change the answer space freely.

【Exercise 4-1】

Exercise 4-1

1. Extract the "definition" and "declaration" of the structure complex from the source code (03-06) and show it.
2. Rewrite "???" on lines 16-18 of the source code (03-06) and execute the basic operations (reference and assignment of members) of the structure.
3. At line 21 of the source code (03-06), add a description that copies the entire structure a to structure b.
4. Show the execution result of the created program.

1)

```
struct complex
{
    float r;
    float i;
};
```

2)

```
a.r =1.1;
a.i =2.2;
printf("complex a = %f + %f i\n",a.r, a.i);
```

3)

```
b = a; // 構造体 a を構造体 b にコピー
```

4)

```
complex a = 1.000000 + 0.000000 i
complex a = 1.100000 + 2.200000 i
complex b = 1.100000 + 2.200000 i
```

【Exercise 4-2】

Exercise 4-2

1. Rewrite the 19th line of the source code (03-07) appropriately as follows: "Use the function `mk_cmplx` to create a complex number of $2.0+5.0i$ and record it in the structure `a`".
2. Show the execution result of the created program.

1)

```
a = mk_cmplx(2.0, 5.0);
```

2)

```
complex a = 1.000000 + 0.000000 i
```

```
complex a = 2.000000 + 5.000000 i
```

【Exercise 4-3】

Exercise 4-3

1. Rewrite "???" in the 18th line of the source code (03-08) as appropriate to match the argument of the function `mk_complex2`.
2. Rewrite "???" on lines 27-28 of the source code (03-08) as appropriate, and execute the structure operation (member assignment) when using a pointer.
3. Show the execution result of the created program.
4. Check the operation when lines 27-28 of source code (03-08) are replaced with the following two lines, compare it with Exercise 4-3-3, and explain the reason for that.

Line 27: `(*c).r = x;`
Line 28: `(*c).i = y;`

1)

```
mk_complex2(&a, 1.0, 2.0);
```

2)

```
c->r = x;  
c->i = y;
```

3)

```
complex a = 1.000000 + 0.000000 i  
complex a = 1.000000 + 2.000000 i
```

4)

Both approaches work the same. The approach in 4-3-2 takes the pointer variable and uses the arrow operator to access the members of the structure directly through the pointer. This approach in 4-3-4 access the members by referring the “the value of” the pointer variable. They are essentially doing the same thing.

【Exercise 4-4】

Exercise 4-4

1. In the source code (03-09) 26-29 line "????", Rewrite appropriately to realize the operation of "assignment and reference to member {x,y,z} (float type decimal value) of member "s" (world type structure) of line type structure "l1".
2. Rewrite the 32nd line of the source code (03-09) as appropriate to realize the process of assigning the address of the line type structure "l1" to the pointer variable "lptr".
3. In the source code (03-09) line 35-37 "????", Rewrite it appropriately so that it is assigned to the member {x,y,z} (float type decimal value) of the line type structure member "s" (world type structure) of the pointer variable "lptr".
4. Show the execution result of the created program.

1)

```
l1.s.x = 1.0;
l1.s.y = 2.0;
l1.s.z = 3.0;
printf("%f %f %f \n", l1.s.x, l1.s.y, l1.s.z);
```

2)

```
lptr = &l1;
```

3)

```
lptr->s.x = 4.0;
lptr->s.y = 5.0;
lptr->s.z = 6.0;
```

4)

```
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
```

【Exercise 4-5】

Exercise 4-5

1. From the data structure specifications shown below, create a program for the structure "polyline" that represents a polygonal line using a linear list (only the structure definition is OK).
2. Show the linear list created in 1.

- Polyline: struct polyline
- Line segment → struct line
- Color → struct color
- Thickness → float
- Pointer to the next line segment → struct polyline
- Color: struct color
- R component → float
- G component → float
- B component → float
- Point: struct point
- X coordinate → float
- Y coordinate → float
- Z coordinate → float
- Line segment: struct line
- Start point → struct point
- End point → struct point

1)

```
#include <stdio.h>

struct color{
    float r;
    float g;
    float b;
};

struct point{
    float x;
    float y;
    float z;
};

struct line{
    struct point start;
    struct point end;
};

struct polyline{
    struct line line;
    struct color color;
    float thickness;
    struct polyline *next;
};
```

2) I don't understand what you meant by "show". From the Japanese version of this assignment, it sounds like you wanted an illustration of the linear list of this polyline structures?

