C Programming

Lab 8: struct and union

Lecturer: *Dr*. Wan-Lei Zhao

Spring Semester 2022

Contact: wlzhao@xmu.edu.cn

Outline

1. struct

2. union

struct (1)

- Define a struct for Complex number (**Compx**)
  - real (**real**) and virtual (**virt**) part
- Define a function to perform multiplication between two complex numbers
- struct Compx multComp(struct Compx d1, struct Compx d2)
  - ....
  - rslt.rl = a.rl*b.rl - a.vt*b.vt;
  - rslt.vt = a.rl*b.vt + b.rl*a.vt;
  - ...
- Define a function "**void printCompx(struct Compx a)**" to print out a complex number
  - It prints like following:
    0.3+0.5i

struct (2): the answer

```c
#include <stdio.h>
struct Compx{
  float rl;
  float vt;
};
struct Compx multComp(struct Compx d1, struct Compx d2)
{
    struct Compx r;
    r.rl = d1.rl*d2.rl-d1.vt*d2.vt;
    r.vt = d1.rl*d2.vt+d2.vt*d1.rl;
    return r;
}
void printComp(struct Compx r)
{
    if(r.vt > 0)
        printf("%f+%fi\n", r.rl, r.vt);
    else if(r.vt < 0)
        printf("%f%fi\n", r.rl, r.vt);
    else
        printf("%f\n", r.rl);
}
```

struct (3): the answer

```c
int main()
{
    struct Compx d1 = {1.2,5.3}, d2 = {1.2,-1.3};
    struct Compx r = multComp(d1, d2);
    printComp(r);
    return 0;
}
```

struct array (1)

- Define a struct named **NoteBook**
  - qq: number, int type
  - name[32]: friend's name, char type
  - phone[16]: phone number, char type
- Define an array (**5 elements**) of NoteBook
  - Input five records
  - Output five records
- Please check the size of your defined struct type

struct array (2)

```c
#include <stdio.h>
struct NoteBook {
  long qq;
  char name[32];
  char phone[16];
};
typedef struct NoteBook QQBook;
void printQQbook();
int main()
{
  printQQbook();
  return 0;
}
```

```c
void printQQbook(){
{
  QQBook persons[4];
  int i = 0;
  for(i = 0; i < 4; i++)
  {
     printf("Name:␣");
     scanf("%s", persons[i].name);
     printf("QQ:␣");
     scanf("%d", &persons[i].qq);
  }
  for(i = 0; i < 4; i++){
  {
    printf("Name:␣%s\n", persons[i].name);
    printf("QQ:␣%d\n", persons[i].qq);
  }
}
```

Outline

1. struct

2. union

union (1)

- Define a union type
  - One **float** number
  - One **short** number
  - One **char** character

  - Use typedef to define type '**DATA**' of above union type
  - Declare variable d1 of type **DATA**

union (2)

```
#include <stdio.h>
union Data {
  float f;
  char c;
  short i;
};
typedef union Data DATA;
void testUnion();
int main()
{
  testUnion();
  return 0;
}
```

```
void testUnion()
{
  DATA d1;
  printf("Size of data: %d\n",
        sizeof(DATA));
  d1.c = 'a';
  printf("%c\n", d1.c);
  d1.f = 3.1415;
  printf("f: %f\n", d1.f);
  printf("d: %d\n", d1.i);
  printf("c: %c\n", d1.c);
  d1.i = 9;
  printf("f: %f\n", d1.f);
  printf("i: %d\n", d1.i);
  printf("c: %c\n", d1.c);
}
```