C语言基础Day8-共用体

一、共用体（联合体）概述

- 联合Union是一个能在同一个存储空间存储不同类型数据的类型
- 联合体所占的内存长度等于其最长成员的长度倍数，也有叫做共用体
- 同一内存段用来存放几种不同类型的成员，但是每一瞬间只有一种起作用
- 共用体变量中起作用的成员是最后一次存放的成员，在存入一个新的成员之后原有的成员的值会被覆盖
- 共用体变量的地址和他的各个成员的地址都是同一地址

多个变量共用同一块内存空间，在同一时刻只有这一种变量起作用。

二、共用体赋值

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define _CRT_SECURE_NO_WARNINGS

// 定义联合体  三种类型共用同一块内存空间
union mi
{
    char a;
    short b;
    int c;
};

int main()
{
    union mi tmp;
    tmp.a = 0x01;// 占用一个字节
    tmp.b = 0x0102;// 占用两个字节
    tmp.c = 0x01020304;// 占用四个字节

    // 所以共用体这块内存空间最后所存放的一定是01020304
    return 0;
}
```

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define _CRT_SECURE_NO_WARNINGS

// 定义联合体  三种类型共用同一块内存空间
union mi
{
```

```
    char a;
    short b;
    int c;
};

int main()
{
    union mi tmp;
    tmp.a = 0x01;// 占用一个字节
    tmp.c = 0x01020304;// 占用四个字节
    tmp.b = 0x0a0b;// 占用两个字节

    // 所以共用体这块内存空间最后所存放的一定是01020a0b
    printf("%x\n",tmp.a);
    printf("%x\n",tmp.b);
    printf("%x\n",tmp.c);
    return 0;
}
```

三、共用体判断大小端

判断计算机是大端存储还是小段存储

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define _CRT_SECURE_NO_WARNINGS

// 定义联合体  三种类型共用同一块内存空间
union mi
{
    short b;
    char buf[2];
};

int main()
{
    union mi tmp;
    tmp.b = 0x0102;//赋值 01 高地址 02 低地址

    if(tmp.buf[0] == 0x01)
    {
        printf("big\n");// tmp.buf[0] 一定存的是低地址  低地址存储的是01
    }
    else
    {
        printf("little\n");// 小端
    }

    printf("%d\n",sizeof(tmp));// 四个字节
```

```
    return 0;
}
```

四、枚举实现布尔类型

枚举：将变量的值一一列举出来，变量的值只限于列举出来的值得范围内。

枚举类型定义： enum 枚举名 { 枚举值表 };

- 在枚举值表中应该列出所有可用值，也称为枚举元素
- 枚举值是常量，不能在程序中用赋值语句再对它进行赋值
- 枚举元素本身由系统定义了一个表示序号的数值从0开始顺序定义为0，1，2，3，...

枚举就是常量

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define _CRT_SECURE_NO_WARNINGS

enum ab{SUN,RAIN,SNOW};// SUN = 0 RAIN = 1 SNOW = 2  初始化 常量

int main()
{
    printf("%d %d %d",SUN,RAIN,SNOW);// 打印 0  1  2



    return 0;
}
```

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define _CRT_SECURE_NO_WARNINGS

// enum ab{SUN,RAIN,SNOW};// SUN = 0 RAIN = 1 SNOW = 2  初始化 常量

enum booll {False,True};// 定义布尔变量 0 1

int main()
{
    printf("%d %d",False,True);// 打印 0  1
    return 0;
}
```

五、typedef 取别名

```c
#include<stdio.h>
#include<stdlib.h>
#define _CRT_SECURE_NO_WARNINGS

typedef int u32;

int main()
{

    int a = 10;
    u32 b = 1;
    printf("%d %d",sizeof(a),sizeof(b));
}
```

```c
#include<stdio.h>
#include<stdlib.h>
#define _CRT_SECURE_NO_WARNINGS

typedef int u32;

struct tt{
    int id;
    int age;
};

typedef struct tt TT;// 取别名

int main()
{

    int a = 10;
    u32 b = 1;
    printf("%d %d",sizeof(a),sizeof(b));

    TT tmp;
    tmp.id = 10;
    tmp.age = 1;

    printf("%d %d",tmp.id,tmp.age);

}
```