C语言qsort函数的使用

- 函数定义

```
void qsort( void *base, size_t num, size_t width, int (__cdecl *compare )
(const void *elem1, const void *elem2 ) );
```

- 参数声明
  - 参数1：base:待排序的数组首地址
  - 参数2：num:待排序的数组中元素的个数
  - 参数3：width:待排序的数组中的元素所占字节数
  - 参数4：int (__cdecl *compare)(const void *elem1, const void *elem2)：函数指针，表明排序规则。
    - elem1,elem2可以理解为待排序的数组中第i个元素与第i+1个元素的地址
      - void*为空指针，可以接受任意类型的地址
      - void*不能进行解引用操作
      - void*不能进行加减法运算
    - 返回值为整形int
      - 数组中第i个元素大于第i+1个元素，函数的返回值为正数；
      - 数组中第i个元素等于第i+1个元素，函数的返回值为0；
      - 数组中第i个元素小于第i+1个元素，函数的返回值为负数；
- 举例：

  1.使用qsort函数对结构体数组进行排序，按照学生的姓名排序

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct Student {
    char name[20];
    int age;
};
int cmp(const void * elem1, const void * elem2)
{
    return strcmp(((struct Student*)elem1)->name, ((struct Student*)elem2)-
>name);
}
void test(struct Student arr[], int sz)
{
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
    qsort(arr, sz, sizeof(arr[0]), cmp);
    printf("\n");
    printf("**********************\n");
    printf("**********************\n");
    printf("**********************\n");
```

```
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
}


int main()
{
    struct Student s[3] = { {"zhangsan",20},{"lisi",30},{"wangwu",10} };
    int sz = sizeof(s) / sizeof(s[0]);
    test(s, sz);
    return 0;
}
```

2. 使用qsort函数对结构体数组进行排序，按照学生的年龄

```
#include<stdio.h>
#include<stdlib.h>

struct Student {
    char name[20];
    int age;
};
int cmp(const void * elem1, const void * elem2)
{
    return ((struct Student*)elem1)->age-((struct Student*)elem2)->age;
}
void test(struct Student arr[], int sz)
{
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
    qsort(arr, sz, sizeof(arr[0]), cmp);
    printf("\n");
    printf("**********************\n");
    printf("**********************\n");
    printf("**********************\n");
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
}


int main()
{
    struct Student s[3] = { {"zhangsan",20},{"lisi",30},{"wangwu",10}
};
    int sz = sizeof(s) / sizeof(s[0]);
```

```
    test(s, sz);
    return 0;
}
```

3.使用qsort函数对整形数组进行排序

```
#include<stdio.h>
#include<stdlib.h>
int cmp_int(const void* elem1, const void* elem2)
{
    return *(int*)elem1 - *(int*)elem2;
}
void test1(int arr[], int sz)
{
    for (int i = 0; i < sz; i++)
    {
        printf("%4d", arr[i]);
    }
    printf("\n");
    printf("*************************\n");
    printf("*************************\n");
    printf("*************************\n");
    qsort(arr, sz, sizeof(arr[0]), cmp_int);
    for (int i = 0; i < sz; i++)
    {
        printf("%4d", arr[i]);
    }
}
int main()
{
    int arr[] = { 1,3,5,2,4,7,9,10,20,23 };
    int sz = sizeof(arr) / sizeof(arr[0]);
    test1(arr, sz);
}
```

- 函数自我实现：

```
void Swap(char* elem1, char* elem2,int width)
{
    for (int i = 0; i < width; i++)
    {
        char temp = *elem1;
        *elem1 = *elem2;
        *elem2 = temp;
        elem1++;
        elem2++;
    }
}
void myqsort(void* base, int sz, int width, int(*cmp)(const void* elem1,
const void* elem2))
{
    for (int i = 0; i < sz - 1; i++)
    {

        for (int j = 0; j < sz - 1 - i; j++)
        {
```

```
            //数组中第j个元素与第j+1个元素的地址
            if (cmp((char*)base + j * width, (char*)base + (j + 1) *
width)>0)
            {
                Swap((char*)base + j * width, (char*)base + (j + 1) *
width,width);
            }
        }
    }
}
```

- 测试代码：

  1.对整形数组进行排序

```
#include<stdio.h>
#include<stdlib.h>
void Swap(char* elem1, char* elem2, int width)
{
    for (int i = 0; i < width; i++)
    {
        char temp = *elem1;
        *elem1 = *elem2;
        *elem2 = temp;
        elem1++;
        elem2++;
    }
}
void myqsort(void* base, int sz, int width, int(*cmp)(const void*
elem1, const void* elem2))
{
    for (int i = 0; i < sz - 1; i++)
    {

        for (int j = 0; j < sz - 1 - i; j++)
        {
            //数组中第j个元素与第j+1个元素的地址
            if (cmp((char*)base + j * width, (char*)base + (j + 1) *
width) > 0)
            {
                Swap((char*)base + j * width, (char*)base + (j + 1) *
width, width);
            }
        }
    }
}
int cmp_int(const void* elem1, const void* elem2)
{
    return *(int*)elem1 - *(int*)elem2;
}
void test1(int arr[], int sz)
{
    for (int i = 0; i < sz; i++)
    {
        printf("%4d", arr[i]);
    }
    printf("\n");
```

```
    printf("************************\n");
    printf("************************\n");
    printf("************************\n");
    myqsort(arr, sz, sizeof(arr[0]), cmp_int);
    for (int i = 0; i < sz; i++)
    {
        printf("%4d", arr[i]);
    }
}
int main()
{
    int arr[] = { 1,3,5,2,4,7,9,10,20,23 };
    int sz = sizeof(arr) / sizeof(arr[0]);
    test1(arr, sz);
}
```

2.对结构体数组进行排序，按照学生年龄

```
#include<stdio.h>
#include<stdlib.h>

void Swap(char* elem1, char* elem2, int width)
{
    for (int i = 0; i < width; i++)
    {
        char temp = *elem1;
        *elem1 = *elem2;
        *elem2 = temp;
        elem1++;
        elem2++;
    }
}
void myqsort(void* base, int sz, int width, int(*cmp)(const void*
elem1, const void* elem2))
{
    for (int i = 0; i < sz - 1; i++)
    {

        for (int j = 0; j < sz - 1 - i; j++)
        {
            //数组中第j个元素与第j+1个元素的地址
            if (cmp((char*)base + j * width, (char*)base + (j + 1) *
width) > 0)
            {
                Swap((char*)base + j * width, (char*)base + (j + 1) *
width, width);
            }
        }
    }
}
struct Student {
    char name[20];
    int age;
};
int cmp(const void * elem1, const void * elem2)
{

```

```
    return ((struct Student*)elem1)->age - ((struct Student*)elem2)-
>age;
}
void test(struct Student arr[], int sz)
{
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
    myqsort(arr, sz, sizeof(arr[0]), cmp);
    printf("\n");
    printf("**********************\n");
    printf("**********************\n");
    printf("**********************\n");
    for (int i = 0; i < sz; i++)
    {
        printf("%d :", i + 1);
        printf("%s  ", arr[i].name);
        printf("%d  ", arr[i].age);
    }
}

int main()
{
    struct Student s[3] = { {"zhangsan",20},{"lisi",30},{"wangwu",10}
};
    int sz = sizeof(s) / sizeof(s[0]);
    test(s, sz);
    return 0;
}
```