基础作业

1什么符号决定了函数的作用域?

- `{}`

2从变量存在的生命周期、变量存储的位置来说明全局变量、静态局部变量和局部变量的区别

-

	生命周期	变量存储的位置
全局变量	从程序运行开始到结束	储存在数据段
静态局部变量	从定义开始到结束	储存在数据段
局部变量	离得最近的{ }	存储在栈中

3对于某一个函数而言，可以定义多少次？声明多少次?

- 一个函数仅可定义一次，可以被声明多次

4对于某个全局变量而言，可以定义多少次，可以使用extern关键字声明多少次?

- 对于全局变量而言，仅可被定义一次
- 可以使用extern多次

5说明静态函数和静态全局变量的特点。总结全局变量和函数在作用域上面的相之处，说一下你对“函数都是外部的“这句话的理解。

- 静态函数和静态全局变量都是从定义开始到程序结束始终都存在的
- 函数都是外部的：函数都是在外部调用函数体/语句处理数据后返回。

6如何定义一个结构体，如何定义一个结构体类型变量，如何定义一个结构体类型的数组和指针

- 定义结构体

```
struct +结构体名+{成员列表};
```

- 定义一个结构体类型的数组

```
typedef struct student {成员列表}Student_t,*pStudent_t;
student_s addr[1024];
```

- 定义一个结构体类型的指针

- ```
  typedef struct student {成员列表}Student_t,*pStudent_t;
  pStudent_t p_new= *pp_head;
  ```

7 什么是结构体变量的对齐

- 在用sizeof函数计算结构体所占的空间时，用最长数据类型成员所占的空间来表示单位变量占的空间。所占空间为：成员个数*最长数据类型成员所占的空间

8 如何遍历一个链表

- 建立一个指针指向链表的头节点，依次移动指针的位置
- ```
  pStudent_t pCur = *ppHead;
  while(pCur)
  {   cout<<pCur.data<<endl;
      pCur = pCur->pNext;
  }
  ```
-

9 熟练掌握结构体指针类型的使用。如何修改一个链表结点指针变量的指向？如何修改一个链表结点指针变量所指向的结点的内容？如何修改一个链表结点指针变量所指向的结点的next指针的指向？

- 修改一个链表结点指针变量的指向？
  - 用箭头的方式修改
  - ```
    pStudent_t p_new = (pStudent_t)calloc(1, sizeof(Student_t));
    pStudent_t pCur = *pp_head;
    pCur->pNext = p_new;
    ```
- 修改一个链表结点指针变量所指向的结点的内容
  - ```
    pCur.data=p_new.data;
    ```
- 如何修改一个链表结点指针变量所指向的结点的next指针的指向

10 实现链表的头插法

- ```
  void listHeadInSert(pStudent_t* pp_head, Student_t** pp_tail, int val)
  {
      pStudent_t p_new = (pStudent_t)calloc(1, sizeof(Student_t));
      p_new->num = val;
      if (*pp_head == NULL) //判空
      {
          *pp_head = p_new;
          *pp_tail = p_new;
      }
      else  //非空则新指针的next指针指向头节点
      {        //新指针为新的头节点
          p_new->pNext = *pp_head;
          *pp_head = p_new;
      }

  ```
  ```

```
16 }
```

-