

20160310学习笔记

函数

静态存储类 和 动态存储类

- 自动的（auto）
- 静态的（static）
- 寄存器的（register）
- 外部的（extern）

别的文件的函数想要调用要加extern

递归

求阶乘

```
1.  int cal(int n){
2.      if(n ==1){
3.          return 1;
4.      } else{
5.          return n*cal(n-1);
6.      }
7.  }
8.  void main(){
9.      int n;
10.     while(fflush(stdin),scanf("%d",&n)!=EOF){
11.         printf("the %d! = %d\n",n,cal(n));
12.     }
13.     system("pause");
14. }
```

冒泡排序

```
1.  func.h
2.  #include <stdio.h>
3.  #include <stdlib.h>
4.  #include <time.h>
5.  #define N 10
6.  void sort_beign(int *);
7.  void sort_printf(int *);
8.  void sort_bubble(int *);
9.  #define SWAP(a,b) {int tmp;\tmp=a;\b=a;\tmp=b;}
```

```
1.  main.c
2.  #include "func.h"
3.  void main(){
4.      int a[N];
5.      sort_beign(a);
6.      sort_print(a);
7.      sort_bubble(a);
8.      sort_print(a);
9.      system("pause");
10. }
```

```
1.  func.c
2.  #include "func.h"
3.
4.  void sort_begin(int *a){
5.      int i;
6.      srand(time(NULL));
7.      for(i=0;i<N;i++){
8.          a[i] = rand()%100;
9.      }
10. }
11.
12. void sort_print(int *a){
13.     int i;
14.     for(i=0;i<N;i++){
15.         printf("%3d",a[i]);
16.     }
17.     printf("\n");
18. }
19.
20. //冒泡排序
21. void sort_bubble(int *a){
22.     int i,j;
23.     for(i=0;i<N-1;i++){
24.         for(j=i+1;j<N;j++){
25.             if(a[i]>a[j]){
26.                 SWAP(a[i],a[j]);
27.             }
28.         }
29.     }
30. }
31.
32. //选择排序，一遍只交换一次
33. void sort_select(int *a){
34.     int i,j;
35.     int min_pos;
36.     for(i=0;i<N;i++){
37.         min_pos = i;
38.         for(j=i+1;j<N;j++){
39.             if(a[min_pos] > a[j]){
40.                 min_pos = j;
41.             }
42.         }
43.         SWAP(a[i],a[min_pos]);
44.     }
```

```
45.     }
46.
47.     //插入排序
48.     void sort_insert(int *a){
49.         int i,j,k,insert_value;
50.         for(i=1;i<N;i++){
51.             for(j=0;j<i;j++){
52.                 if(a[i]<a[j]){
53.                     insert_value = a[i];
54.                     for(k=i;k>j;k--){
55.                         a[k] = a[k-1];
56.                     }
57.                     a[j] = insert_value;
58.                     break;
59.                 }
60.             }
61.         }
62.     }
63.
64.     int compare(const void*left,const void*right){
65.         int *pl =(int *)left;
66.         int *pr =(int *)right;
67.         if(*pl>*pr){
68.             return 1;
69.         }else if(*pl<*pr){
70.             return -1;
71.         }else{
72.             return 0;
73.         }
```

结构体

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  void main(){
4.      struct student{
5.          int num;
6.          char name[20];
7.          char sex;
8.          int age;
9.          float score;
10.         char addr[20];
11.     };
12.     struct student s1,s2;//大小并不是56, 存在边界对其
13.     s1.num = 1000;
14.     strcpy(s1.name, "liming");
15.     s1.sex = 'M';
16.     s1.age = 24;
17.     s1.score = 90.3;
18.     strcpy(s1.addr, "shenzhen");
19.     system("pause");
20.     //memcpy(&s2,&s1,sizeof(s1));
21.     scanf("%d%s%d%c%f%s",&s1.num,s1.name,&s1.age,&s1.sex,&s1.score,s1.addr);
22. }
23.
24.
```