

## 复杂编程题(40分)

## 1. (10分)

编程统计候选人的得票数。设有3个候选人zhang、li、wang（候选人姓名不区分大小写），10个选民，选民每次输入一个得票的候选人的名字，若选民输入错误候选人姓名，则按废票处理。选民投票结束后程序自动显示各候选人的得票结果和废票信息。要求用结构体数组candidate表示3个候选人的姓名和得票结果。

```
#include <stdio.h>
#include <string.h>
#define NUM_ELECTORATE 10
#define NUM_CANDIDATE 3
struct candidate
{
    char name[20];
    int count;
} candidate[3] = {"li", 0, "zhang", 0, "wang", 0};
int main()
{
    int i, j, flag = 1, wrong = 0;
    char name[20];
    for (i = 1; i <= NUM_ELECTORATE; i++)
    {
        printf("Input vote %d:", i);
        scanf("%s", name);
        strlwr(name);          /* 将name中的字符全部变成小写字母 */
        flag = 1;
        for (j = 0; j < NUM_CANDIDATE; j++)
        {
            if (strcmp(name, candidate[j].name) == 0)
            {
                candidate[j].count++;
                flag = 0;
            }
        }
        if (flag)
        {
            wrong++; /* 废票计数 */
            flag = 0;
        }
    }
    printf("Election results:\n");
    for (i = 0; i < NUM_CANDIDATE; i++)
    {
        printf("%8s:%d\n", candidate[i].name, candidate[i].count);
    }
    printf("Wrong election:%d\n", wrong);
    return 0;
}
```

## 2. (10分)

请按给定的函数原型，编写程序判断输入的字符串是否为“回文”。所谓“回文”，是指顺读和倒读都一样的字符串。如：“level”、“ABCCBA”都是回文。如是回文输出这个字符串，否则输出“No!”。要求：在主函数读入字符串，判断是否是回文调用Judger()函数实现。若是回文返回值为1，否则返回值为0。Judger函数原型：int Judger( char a[ ]);

\*\*输入格式要求：无

\*\*输出格式要求：“No!” （不是回文） 说明：是回文则输出串本身

程序运行的输入输出样例1：

输入：

abc

输出：

No!

程序运行的输入输出样例2：

输入：

12321

输出：

12321

```
#include <stdio.h>
#include <string.h>
int Judger( char a[ ]);
int main()
{
    char a[80];
    int flag;
    gets(a);
    flag = Judger(a);
    if (flag == 1)
    {
        puts(a);
    }
    else
    {
        printf("No!");
    }
    return 0;
}
int Judger( char a[ ])
{
    char *pStar, *pEnd;
    int i, j, flag = 1;
    j = strlen(a);
    pStar = &a[0];
    pEnd = &a[j - 1];
    for (i = 0; i < (j / 2); i++)
    {
        if (*pStar != *pEnd)
        {
            flag = 0;
        }
        pStar++;
        pEnd--;
    }
    return flag;
}
```

3. (10分) 请按如下函数原型编程实现将字符数组中的字符串逆序存放。要求在主函数读入字符串，并在主函数中输出逆序存放后的字符串。

```
void inverse(char *p);
```

要求：

- 1) 输入的字符串，应包含字母，数字以及空格字符。
- 2) 字符串最大长度为80。
- 3) 没有输入输出提示信息。

请按如下框架编程：

```
1  #include<stdio.h>
2  void inverse(char *p);
3  int main()
4  {
5
6      return 0;
7  }
8  void inverse(char *p)
9  {
10
11 }
```

```
#include<stdio.h>
void inverse(char *p);
int main()
{
    char a[80];
    gets(a); //1
    inverse(a); //1
    puts(a); //1
}
void inverse(char *p)
{
    char *q;
    char t;
    q = p; //1
    while (*q != '\0') //1
    {
        q++; //1
    }
    for (q--; p < q; p++, q--) //2
    {
        t = *p; //1
        *p = *q; //1
        *q = t; //1
    }
}
```

4. (10分) 给定如下定义：

```
1  struct date_rec
2  {
3      int day ;
4      int month ;
5      int year ;
6  };
```

写一个函数接收两个日期，如果两个日期相同，则返回0，如果第一个日期晚于第二个日期，则返回1，如果第一个日期早于第二个日期，则返回-1。并编写main函数测试它。

\*\*输入格式要求: "%d%d%d" 提示信息: "请输入当前日期(年 月 日): "

\*\*输出格式要求: "当前日期: %d年%d月%d日! \n" "第一个日期晚于第二个日期!" "第一个日期早于第二个日期!" "两个日期相同!"

程序运行示例如下：

请输入当前日期(年 月 日): 2012 9 10  
请输入当前日期(年 月 日): 2013 7 10

当前日期: 2012年9月10日! <== 输出  
当前日期: 2013年7月10日!  
第一个日期早于第二个日期!

资料由公众号【工大喵】收集整理并免费分享

```

#include <stdio.h>

struct date_rec
{
    int day;
    int month;
    int year;
};

struct date_rec date1, date2;

void input_date(struct date_rec *current_date)
{
    printf("请输入当前日期 (年 月 日) : ");
    scanf("%d%d%d", &current_date->year, &current_date->month, &current_date->day);
}

int compare_date(struct date_rec *date1, struct date_rec *date2)
{
    if (date1->year < date2->year)
    {
        return -1;
    }
    else if (date1->year > date2->year)
    {
        return 1;
    }

    if (date1->month < date2->month)
    {
        return -1;
    }
    else if (date1->month > date2->month)
    {
        return 1;
    }

    if (date1->day < date2->day)
    {
        return -1;
    }
    else if (date1->day > date2->day)
    {
        return 1;
    }

    return 0;
}

void output_date(struct date_rec *current_date)
{
    printf("当前日期: %d年%d月%d日! \n", current_date->year,
        current_date->month, current_date->day);
}

int main()
{
    input_date(&date1);
    input_date(&date2);

    output_date(&date1);
    output_date(&date2);

    switch (compare_date(&date1, &date2))
    {
        case 1:
            printf("第一个日期晚于第二个日期!");
            break;
        case -1:
            printf("第一个日期早于第二个日期!");
            break;
        case 0:
            printf("两个日期相同!");
    }
}

```

```
}  
  
    return 0;  
}
```

---

@2016 HIT SSE