厦门大学《C 语言程序设计》课程
期末试题·答案

考试日期：2013.01　　信息学院自律督导部整理

一、 写出下列程序段的运行结果（40 分）

1、（4 分）

```
char a='D'-'A',b='\010',c,d;
c = ++a || b++;
d = a-- && b--;
printf("%d,%d,%d,%d",a,b,c,d);
```

输出:3,7,1,1

2、（4 分）

```
char a=0x123FFF;
printf("%d,",a);
a=127;
printf("%d",++a);
```

输出:-1,-128

3、（4 分）

```
int a=5,y=0;
int main()
{
  while(a=5)
  {
    a+=5;
    y+=a;
    printf("%d,%d,", a, y);
    if (y>18) break;
  }
  return 0;
}
```

输出:10,10,10,20

4、（4 分）

```
int i=1, j=2, k=4;
printf("%f,", i-(float)j/k);
printf("%d,", j>1<k);
k+=5/10*i ;
printf("%d,",k);
printf("%d", (j=1)?2:0);
```

输出:0.500000,1,4,2

5、（4 分）

```
int a[][2]={(1,2),(3,4)};
printf("%d",a[0][1]);
```

输出:4

6、（4 分）

```
int ave(int x, int y)
{
  return (x+y)/2;
}
int main()
{
  int a=1, b=2, c=3;
  printf("%d\n",
      ave(ave(a,b), ave(b,c)));
  return 0;
}
```

输出:1(\n)

7、（4 分）

```
void exc(int x, int *y)
{
  int t;
  t=x; x=*y; *y=t;
}
int main()
{
  int a=5,b=8;
  exc(a,&b);
  printf("a=%d,b=%d",a,b);
```

```
   return 0;
}
```
输出：a=5,b=5

8、　　(4 分)
```
char str[10]="believe";
char *p=str;
*(str+5)=0;
puts(2+p);
```
输出：lie

9、 (4 分)
```
struct pt
{
   int x;   int y;
} p[2]={1,3,5,7};
printf("%.3f",
    p[1].y/(float)p[0].y
     + p[1].x/p[0].x);
```
输出：7.333

10、　　(4 分)
```
FILE *fp;
char c=49;
int d=50, e;
fp=fopen("a.tmp","w");
fprintf(fp,"%c%d", c, d);
fclose(fp);
fp=fopen("a.tmp","r");
fscanf(fp,"%d",&e);
printf("%d\n",e);
```
输出：150（\n）

二、 改错题（20 分）

以下程序实现：从键盘输入若干个学生的成绩数据，将这些数据存到磁盘文件上，并求这些学生的最高成绩。程序 21 行前存在 4 个错误，其余部分存在若干错误，指出错误所在位置并改正。

```
#define SIZE 80;                    /*第 1 行*/去掉分号;
#include <stdio.h>                  /*第 2 行*/
struct STU                          /*第 3 行*/
{                                   /*第 4 行*/
  char name[10];                    /*第 5 行*/
  double score;                     /*第 6 行*/
}                                   /*第 7 行*/ }后加分号；
void save(struct STU stud[SIZE])    /*第 8 行*/
{                                   /*第 9 行*/
   FILE   fp;                       /*第 10 行*/ *fp
   int i;                           /*第 11 行*/
   if((fp=fopen("stu-list","wb"))==NULL)     /*第 12 行*/
   {                                /*第 13 行*/
      printf("cannot open file\n");  /*第 14 行*/
      return;                       /*第 15 行*/
   }                                /*第 16 行*/
   for(i=0;i<SIZE;i++)              /*第 17 行*/
    if(fwrite(stud[i],sizeof(struct STU),1,fp)!=1)     /*  第  18  行
*/&stud[i]
      printf("file write error\n");     /*第 19 行*/
   fclose(fp);                          /*第 20 行*/
}                                   /*第 21 行*/
int   main()                        /*第 22 行*/
{                                   /*第 23 行*/
   struct STU stud[SIZE];           /*第 24 行*/
   double maxscore;                 /*第 25 行*/函数声明 double max(struct
STU stud[SIZE]);
   for(i=0;i<SIZE;i++)              /*第 26 行*/
```

```
    {                                   /*第 27 行*/
      scanf("%lf%s ",&stud[i].score,&stud[i].name);  /*第  28  行*/去掉第二个&
    }                                   /*第 29 行*/
    save(struct stud);                  /*第 30 行*/去掉 struct
    maxscore = max(stud[SIZE]);     /*第 31 行*/去掉[SIZE]
    printf("Maxscore is :%8.2lf\n", maxscore);   /*第 32 行*/
     return 0;                              /*第 33 行*/
}                                       /*第 34 行*/
double max(struct STU stud[SIZE])/*第 35 行*/
{                                       /*第 36 行*/
    double tmp;                         /*第 37 行*/tmp=0;
    int i;                              /*第 38 行*/
    for(i=0;i<SIZE;i++)                 /*第 39 行*/
      if(tmp> stud[i].score)            /*第 40 行*/>改<
      {                                    /*第 41 行*/
        tmp= stud[i].score;               /*第 42 行*/
      }                                    /*第 43 行*/
    return tmp;                          /*第 44 行*/
}                                       /*第 45 行*/
```

三、编程题（40 分）　　注意：程序中请添加必要的注释

1、编程实现以下功能：输入星期几的第一个字母来判断输入的是星期几，如果第一个字母一样，则要求用户继续输入第二个字母。程序要求可以进行多轮判断，直到输入为字符'#'为止。其中输入的星期的首字母必须大写，其余字母小写。如果输入的字母不合法，则输出：data error。（12 分）

（一周 7 天是 Monday、Tuesday、Wednesday、Thursday、Friday、Saturday、Sunday）

解：本题看似简单，其实难度不小，稍不留神就会出错，希望读者可以先自己试着编一遍，体会一下这道题的陷阱

```
#include<stdio.h>
int main()
{
char c[2],a;//定义三个字符，前两个用来储存字母，第三个用来吸收“回车”字符
    scanf("%c",&c[0]);
    a=getchar();//吸收输入完 c[0]后的 “回车”
    while(c[0]!='#')
    {if(c[0]=='M')
        printf("Monday\n");
    else
    {
        if(c[0]=='W')
            printf("Wednesday\n");
        else
        {
            if(c[0]=='F')
                printf("Friday\n");
            else
            {
                if(c[0]=='T')
                {
```

```
        printf("plaese input the second letter:");
        scanf("%c",&c[1]);
        a=getchar();   //也是吸收回车字符
        if(c[1]=='u')
            printf("Tuesday\n");
        else
            if(c[1]=='h')
                printf("Thursday\n");
        else
            printf("date error\n");
    }
    else
    {
        if(c[0]=='S')
        {
            printf("plaese input the second letter:");
            scanf("%c",&c[1]);
            a=getchar();
            if(c[1]=='a')
                printf("Saturday\n");
            else
                if(c[1]=='u')
                    printf("Sunday\n");
```

```
                else
                    printf("date error\n");
            }
            else
                printf("data error\n");
        }
    }
}
    scanf("%c",&c[0]);
    a=getchar();
    }
    return 0;
}
```

2、编写函数：void LRShift(int iaElems[], int iArrayLen, int iCount);

该函数将给定数组iaElems的内容循环右移iCount次。其中，iArrayLen为数组iaElems中包含的元素个数。例如：设数组a的内容为{20, 13, 1, 14}，当调用LRShift(a, 4, 1)时，该数组内容将变为{14, 20, 13, 1}；而当调用LRShift(a, 4, 7)时，该数组内容将变为{13, 1, 14, 20}。（14分）

解：蓝色部分为所编写的函数，蓝色红色部分合在一起可以运行此程序

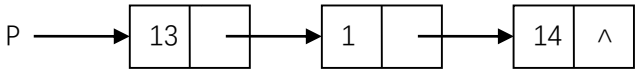
考虑到要进行数组的循环转换，而链表可以通过更改首地址的办法来方便的进行循环转换，笔者也采取了链表法。

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

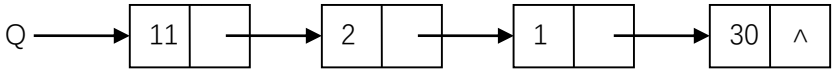
```
#define len sizeof(struct p)
struct p
{
    int a;
    struct p *next;
    struct  p  *before;                                    //向右移动等价于头指针向左移动  所以需要双向链表
};
void LRShift(int iaElems[], int iArrayLen, int iCount)
{
    struct p *p1,*p2,*head;
    int i;
    for(i=0;i<iArrayLen;i++)
    {
        p1=(struct p*)malloc(len);
        p1->a=iaElems[i];
        if(i==0)
            head=p1;
        else
        {
            p1->before=p2;
            p2->next=p1;
        }
        p2=p1;
    }
    p2->next=head;
    head->before=p2; //到此链表建立完成  是一个循环的双向链表，链表尾的next指向头  头的before指向尾
    while(iCount>0)
    {
        head=head->before;
        iCount--;   //将头指针进行向左移动
    }

    for(i=0;i<iArrayLen;i++)
    {
        iaElems[i]=head->a;       //把头指针移动好了的链表元素重新输入进数组中
        printf("%d ",iaElems[i]);
        head=head->next;
    }
    return;

}
int main()
{
    int a[1000],n,m,i;
```

```
    scanf("%d %d",&n,&m);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    LRShift(a,n,m);
    return 0;
}
```

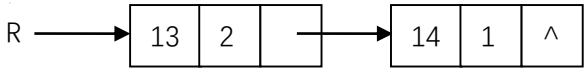
3、用链表存储一组不重复的整数，如13、1、14三个数被存储到链表P：



编写函数SUM(P, Q, n)，参数P和Q分别指向两个链表（表示两组数），函数返回一个新链表R（注意R和P、Q的结点结构不同），表示从P、Q两组数中各取一个数，相加之和等于n的一个组合。例如当n为15，假设Q如下：



执行SUM(P, Q, 15)，结果如下（结点在链表中的排列次序任意，下图只是一例）：



要求：（1）写出P、R链表结点定义；（2分）

（2）编写完成函数SUM，实现题目要求的功能。（12分）

解：本题考链表的合并，较难，需要灵活掌握链表的建立

蓝色字体为P、R链表结点定义 紫色字体为SUM函数 紫红蓝加在一起可以运行此程序 例：输入 15（就是n）

13 1 14 0（以0作为链表结束）

11 2 1 30 0

输出： 13,2 14,1

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

```
#define k1 sizeof(struct l1)
#define k2 sizeof(struct l2)
struct l1
{
 int n;
 struct l1 *next;
};
struct l2
{
 int n;
 int m;
 struct l2 *next;
};
struct l1 *creat()
{
 struct l1 *p1,*p2,*head=NULL;
 int n=0;
 p1=p2=(struct l1*)malloc(k1);
 scanf("%d",&p1->n);
 while(p1->n!=0)
 {
     n++;
     if(n==1)
```

```
        head=p1;
    else
        p2->next=p1;
    p2=p1;
    p1=(struct l1*)malloc(k1);
    scanf("%d",&p1->n);
}
p2->next=NULL;
return(head);
}
struct l2 *sum(struct l1 *p,struct l1 *q,int k)
{
struct l2 *p1,*p2,*head=NULL;
int l=0;
struct l1 *qn;
qn=q;
p1=p2=(struct l2*)malloc(k2);
while(p!=NULL)
{
    while(q!=NULL)
    {
        if(p->n+q->n==k)
        {
```

```
            p1->n=p->n;p1->m=q->n;
            l++;
            if(l==1)
                head=p1;
            else
                p2->next=p1;
            p2=p1;
            p1=(struct l2*)malloc(k2);
        }
        q=q->next;
    }
    p=p->next;q=qn;
}
p2->next=NULL;
return(head);
}

void output(struct l2 *r)
{
while(r!=NULL)
{
    printf("%d,%d ",r->n,r->m);
    r=r->next;
```

```
}
printf("\n");
}


int main()
{
struct l1 *p,*q;
struct l2 *r;
int n;
scanf("%d",&n);
p=creat();
q=creat();
r=sum(p,q,n);
output(r);
return 0;
}
```