

复旦大学科学技术试验班

2012-2013 学年第一学期

《程序设计》期末考试试卷

A 卷 共 7 页

课程代码: COMP120006

考试形式: 开卷 闭卷

2013 年 1 月

(本试卷答卷时间为 120 分钟, 答案必须写在答题卷上, 做在草稿纸或试卷上无效)

(本课程成绩总分为 100 分: 平时成绩满分为 30 分, 本次期末考试满分为 70 分)

学号 _____ 姓名 _____

题号	一	二	三	四	五	试卷总分
满分	10	6	10	30	14	70

一、基本知识题 (10 分)

1.1. 已知定义: int a=0; 请指出以下不会产生死循环的控制结构。

- A. for (; ;) if(a) break;
- B. for (; ; a=0) if(a) break;
- C. for (; ;) if(a) continue;
- D. for (; a=0;) if(a) break;

1.2. 请指出正确描述实参和形参关系的命题。

- A. 如果实参是数组名, 则通过函数调用向函数传送数组的值
- B. 如果实参是数组名, 则通过函数调用向函数传送数组的首地址
- C. 如果实参是数组名, 则通过函数调用向函数传送数组的值
- D. 如果实参是数组名, 则通过函数调用向函数传送数组的值

1.3. 假定在 32 位计算机中, sizeof(long)=4, sizeof(short)=2。根据以下定义语句, 请计算 sizeof(struct x) 的值。

```
struct x
{
    long      english;
    short     * cost[5];
    struct x * next;
};
```

- A. 16
- B. 28
- C. 32
- D. 13

1.4. 已知定义: char *p[10], 请指出该语句的作用。

- A. 定义了一个有 10 个元素的数组, 每个元素指向一个字符
- B. 定义了一个用以指向长度为 10 的字符串的指针变量
- C. 定义了一个有 10 个字符型指针的一维数组
- D. 定义了一个有 10 个元素的数组, 每个元素存放一个字符串。

1.5. 请指出以下错误的命题。

- A. 结构的成员变量与结构变量可以同名

- B. 形式参数都是局部变量（自动变量）
- C. 定义函数类型为 void 表示该函数应该返回一个数据
- D. 函数的传值调用是指实在参数向形式参数单向传递（复制）数值

二、程序运行题（6分）

2.1. (2分) 请写出以下程序的运行结果。

```
void check(int a,int mark,int bit)
{
    int i=0,key=0,b;
    b = (a >> bit) & mark;
    for( ; i<8; i++, b>>=1)
        if(b % 2)
            key++;
    printf("%d ",key);
}
void main()
{
    check(0X8706,0xFD,8);
    check(0XF4AF,0xF7,4);
}
```

2.2. (2分) 如果一下 sub 函数中的输出为“1 2 4 0 4”，请写出 main 函数中的输出。

```
void sub(char *dot,char line[],char n)
{
    char random;
    random = rand() % n;
    *dot = line[random];
    printf("%d ",random);
}
void main()
{
    char i, n, line[10]={‘F’, ‘U’, ‘D’, ‘A’, ‘N’};
    for(i=0,n=5; i<n; i++)
        sub(&line[i+5],line,n);
    for( ; i<n+5; i++)
        printf("%c", line[i]);
}
```

2.3. (2分) 请写出以下程序的运行结果。

```
#include <stdio.h>
void sub(int *a, int n, int k)
{
    if(k <= n)
        sub(a, n/2, 2*k);
    *a += k;
}
void main()
{
    int x=0;
    sub(&x, 5, 1);
    printf("%d\n", x);
}
```

三、阅读理解题 (10 分)

3.1. (2 分) 请写出应输入_____处的内容，以实现该程序统计文件中行数的功能。

```
#include <stdio.h>
#include <string.h>
void main()
{
    FILE *fp;
    int count=0, ch;
    if((fp = 3.1.1) == NULL )
    {
        printf("Can not open file: %s \n", "data.txt");
        exit 0;
    }
    while((ch = fgetc(fp)) != EOF)
        if(3.1.2)
            count++;
    fclose(fp);
    printf("%d\n",count);
}
```

3.2. (4 分) 以下程序运行的结果应该为“a=5, b=3”，请在不增删语句的情况下指明程序中的第几行有错误，并且写出正确的语句。

```
/* 第 1 行 */ void swap(short *x, short *y)
{
    /* 第 2 行 */     short *z;
    /* 第 3 行 */     z = *x;
    /* 第 4 行 */     *x = *y;
    /* 第 5 行 */     *y = z;
}
void main()
{
    /* 第 6 行 */     short a=3, b=5;
    /* 第 7 行 */     swap(a, b);
    /* 第 8 行 */     printf("a=%d, b=%d\n", a, b);
}
```

3.3. (4 分) 已知结构类型为含有双向指针的表元（结点）数据成员 n 的数值以及由指针 now 和 node 所指向的链接关系（图 a），请应填入_____处的程序语句，实现以下操作：将 node 指向的表元插到 now 指向的表元之前，构成图 b 中表元的双向链接。

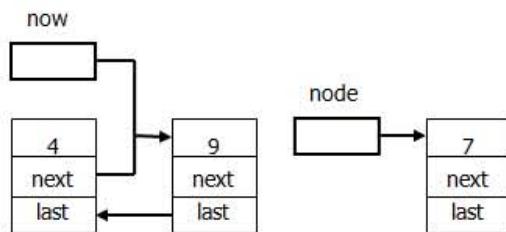


图 a 操作前的链接关系

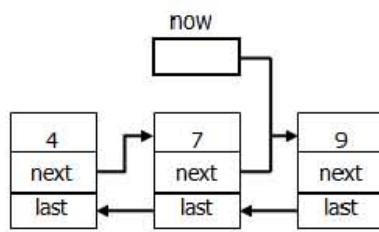


图 b 操作后的链接关系

```
node->next = now;
node->last = now->last;
3.3.1
3.3.2
```

四、编程填空题（30 分）

请在以下试题中写出应填入_____处的字句，以完成各题程序所需要的功能。

4.1. (10 分) 选择排序 (Selection Sort) 是一种基本的排序方法，算法描述为：

假设待排序的数据序列为 $a_0, a_1, a_2, \dots, a_{n-1}$ ，依次对 $i=0, 1, \dots, n-2$ 分别执行如下步骤：
在待排序的数据序列 $a_0, a_1, a_2, \dots, a_{n-1}$ 中选择一个数值最小的项 a_k ，然后将 a_k 与 a_i 交换，直到整个数据序列完成排序。

【C 程序】

```
short *input(short *n)
{
    short i, *a;
    printf("请输入待排序的数据个数和各个数据: ");
    scanf(____4.1.1____); /* 读入 n */
    a = ____4.1.2____;
    for(____4.1.3____) /* 读入待排序的数据 */
        scanf("%d", &a[i]);
    return(a);
}
void main()
{
    short *a, n, i, k, min, tmp;
    a = input(&n); /* 读取 n 个待排序的数据 */
    for(i=0; i<n-1; ++i)
    {
        for(min=i, k=i+1; k____4.1.4____; ++k)
            if(____4.1.5____)
                min=k; /* 确定数值最小的项 a[k] */
        tmp = a[i]; a[i] = a[min]; a[min] = tmp;
    }
    for(i=0; i<n; i++)
        printf("%d ", a[i]);
    putchar('\n');
}
```

4.2. (6 分) 读入一个不超过 6 位的正整数。要求按逆向打印各位数字，求出它是几位数并打印。

例如读入数为 3210，应该输出以下格式的信息：

逆向数是：0123，数据位数：4

要求打印逆向数的每一位时，在 `printf` 语句中使用%c 的格式描述。

【C 程序】

```
#include <stdio.h>
void main()
{
    long a; /* 不超过 6 位的正整数 */
    int k; /* 当前位的一位数 */
    int bit; /* 数据 a 的位数 */
    printf("请输入一个不超过 6 位的正整数: ");
    scanf("%ld", &a);
    printf("逆向数是: ");
    for(____4.2.1____)
    {
        k=____4.2.2____;
        bit++;
        printf(____4.2.3____);
    }
}
```

```
        printf("数据位数: %d\n", bit);
    }
```

- 4.3. (14 分) 假定已将有关学生信息存入一个有哨兵(前置表元)的链表(链表的首指针为 `head->next`)。其中记录着每位学生的学号和绩点。现需要根据给定的优秀绩点的下限(`score`)编程确定获得奖学金的学生名单。

存储学生信息的链表，采用以下结构定义：

```
struct stud_info
{
    int num;           /* 学生学号(有效值为四位正整数) */
    float score;       /* 学生绩点(有效值为 2.00 到 4.00) */
    struct stud_info *next;
};

typedef struct stud_info S;
```

- 1) 请完成以下 `getStud` 函数的编写，只要链表非空，`getStud` 函数将从链表中摘取一个表元(删除结点)，并且返回该学生的表元指针，其条件为学生绩点大于等于形参 `score`(优秀绩点的下限)。若链表中不存在这样的学生，返回 `NULL`。(8 分)

【C 程序】

```
S *getStud(S *head, float score)
{
    S * stud;           /* 获得奖学金学生的表元指针 */
    for(; head->next; head=head->next)
        if( _____ 4.3.1 _____ )
    {
        stud = _____ 4.3.2 _____ ;
        _____ 4.3.3 _____ ;
        _____ 4.3.4 _____ ;
    }
    return NULL;
}
```

- 2) 假定以上编写的 `getStud` 函数是正确的，请完成以下 `makeAward` 函数的编写。

根据形参 `head` 和 `score`，通过调用 `getStud` 函数，从学生链表中获得所有符合奖学金条件的学生信息，存入一个以 `award` 为首指针的链表中。(6 分)

获取所有获奖学金的学生后，输出 `award` 链表中他们的学号和绩点，输出格式示例如下：

```
stud: 1234, score: 3.68
```

【C 程序】

```
void makeAward(S *head, float score)
{ /* 形参 score 是符合奖学金条件的优秀绩点下限 */
    S * stud;           /* 获得奖学金学生的表元指针 */
    S * award=NULL;     /* 获得奖学金学生链表的首指针 */
    while( _____ 4.3.5 _____ )
    {
        stud->next = _____ 4.3.6 _____ ;
        _____ 4.3.7 _____ ;
    }
    for(stud=award; stud; stud=stud->next)
        printf("stud: %d, score: %g\n", stud->num, stud->score);
}
```

调用 `makeAward` 的语句：

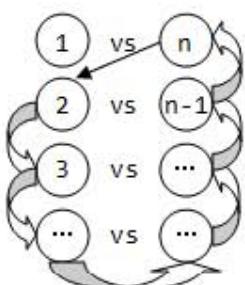
```
makeAward(head, score);
```

五、算法编程题（14分）

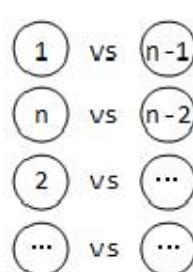
5.1. (6分) 足球等体育比赛常用的一种称为蛇行算法的循环赛制编排方法。如果有 n 个（偶数个）参赛队，在各轮赛程中，1 号队享有优先权，起位置不变，而 2 号参赛队到 n 号参赛队按逆时针循环。

以下为图示的蛇行编排方法，如果当前轮的对阵形式为 1 号队对阵 n 号队，2 号队对阵 $n-1$ 号队等等。通过蛇行算法的编排，在下一轮中，1 号队不变，其他队按逆时针移动，成为 1 号队对阵 $n-1$ 号队， n 号队对阵 $n-2$ 号队等等。

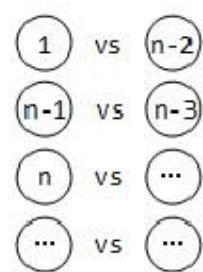
第 1 轮的对阵形势



第 2 轮的对阵形势



第 3 轮的对阵形势



例如，有 6 个队参赛时，输入为第 1 轮（当前轮）的赛程：

```
6          /* 参赛队数（对阵场数等于参赛队数的二分之一） */
1 6        /* 第 1 场对阵形势：1 号队 vs 6 号队 */
2 5        /* 第 2 场对阵形势：2 号队 vs 5 号队 */
3 4        /* 第 3 场对阵形势：3 号队 vs 4 号队 */
```

经过函数 `snake()` 的编排，输出第 2 轮（下一轮）的赛程：

```
1 5        /* 第 1 场对阵形势：1 号队 vs 5 号队 */
6 4        /* 第 2 场对阵形势：6 号队 vs 4 号队 */
2 3        /* 第 3 场对阵形势：2 号队 vs 3 号队 */
```

以下程序中，`main` 函数从键盘读入参赛队数 n ，以及当前轮的赛程，存入二维数组 `turn` 中，通过函数 `snake()` 的编排，将获得下一轮的赛程并输出，请编写函数 `snake()`。

```
#include <stdio.h>
#include <stdlib.h>
void snake(int turn[][2], int n)
{
    int i, key;
    key = turn[n-1][0];
    /* 请在此编写 snake() 所需的程序 */

    turn[n-1][1] = key;
}
void main()
{
    int **turn, n, k;
    printf("请输入参赛队数 n ");
    scanf("%d", &n);
    n = (n + 1) / 2;      /* 对阵场数等于参赛队数的二分之一 */
    for(k=0; k<n; k++)
        scanf("%d%d", &turn[k][0], &turn[k][1]);
    snake(turn, n);
    for(k=0; k<n; k++)
        printf("%d %d\n", turn[k][0], turn[k][1]);
}
```

5.2. (8 分) 寻找字符串中相同字符部分。

已知字符串 `string` 和字符串 `sub`, 寻找字符串 `string` 中含有的字符串 `sub` 部分, 若找到, 返回字符串 `string` 第几个字符最先开始出现字符串 `sub`, 若没找到, 返回-1。

例如:

<code>*string = "babcacbcab" , *sub = "abc"</code>	则 <code>return 1;</code>
<code>*string = "bcbccaaabca" , *sub = "abc"</code>	则 <code>return 6;</code>
<code>*string = "babccabcab" , *sub = "abc"</code>	则 <code>return -1;</code>

若字符串 `string` 或 `sub` 本身是空字符, 则报错。

函数模型: `int substring(*string, *sub)`

```
{  
    /* 请在此编写 substring )所需的程序 */  
}
```

-----完-----

(
装
订
线
内
不
要
答
题
)