**浙江大学2014–2015学年春夏学期**

**《程序设计专题》课程期末考试试卷**

课程号： 211G0260 ，开课学院： 计算机学院\_\_

考试试卷：√A卷、B卷（请在选定项上打√）

考试形式：√闭、开卷（请在选定项上打√），允许带 ∕入场

考试日期： 2015 年 07 月 11 日,考试时间： 120 分钟

**诚信考试，沉着应考，杜绝违纪.**

**考生姓名： 学号： 所属院系： \_**

**(注意：答题内容必须写在答题卷上，写在本试题卷上无效)**

试题一、单选题（每小题2分，共30分）

1. 关于函数调用的说法，错误的是\_\_\_\_\_\_\_.

|  |
| --- |
| A. 递归函数和普通函数一样，都可以使用全局变量。 |
| B．在递归函数中，静态的局部变量只创建一次。 |
| C．嵌套调用函数时，先调用的函数先返回，后调用的函数后返回。 |
| D．递归调用的层数不是无限制的。 |

1. 在最坏情况下，二分查找算法的时间复杂度是\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．O( *1* ) | B．O( *N* ) |
| C．O(log(*N*) ) | D．O( *N*2 ) |

1. 对于以下结构定义，++p->str中的++加在\_\_\_\_\_\_\_.

struct {

intlen;

char \*str;

} \*p;

|  |  |  |  |
| --- | --- | --- | --- |
| A． 指针p上 | B．指针str上 | C．str指的内容上 | D．语法错误 |

1. 以下正确的描述是\_\_\_\_\_\_\_.

|  |
| --- |
| A. 预处理指令只能位于C源程序文件的首部。 |
| B．凡是C源程序中行首以“#”标识的控制行都是预处理指令。 |
| C．C语言的编译预处理就是对源程序进行初步的语法检查。 |
| D．C语言的预处理功能是指完成宏替换和包含文件的调用。 |

1. 在VESA\_800x600x24bit图形模式下，要使数组a刚好能够容纳整个屏幕的图像信息，则a应该定义为\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．char a[800\*600]; | B．char a[800][600][24]; |
| C．struct{char c[3];}a[800][600]; | D．long a[800][600]; |

1. 以下程序段的算法时间复杂度是\_\_\_\_\_\_\_.

for (i=1; i<=n; i++)

x++;

for (i=1; i<=n; i++)

    　for (j=1; j<=n; j++)

          x++;

|  |  |
| --- | --- |
| A．O( *N* ) | B．O(*N*2) |
| C．O( 2*N*) | D．O( *N*3 ) |

1. 假定已定义变量int key;当要读取键盘上的Home或End键并赋值给key时，则正确的写法是\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．key = getchar(); | B．scanf("%d", &key); |
| C．key = fgetc(stdin); | D．key = bioskey(0); |

1. F(x)是一个带参数的宏，欲在宏替换后达到1+x2的计算效果。为了保证F(a+b)\*F(c+d)对于任何取值的浮点数变量a、b、c、d都结果正确，那么该宏正确的定义方式是\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．#define F(x) x\*x+1 | B．#define F(x) (x)\*(x)+1 |
| C．#define F(x) ((x)\*(x)+1) | D．#define F(x) (x\*x+1) |

1. 以下关于全局变量和外部变量的说法，不正确的是\_\_\_\_\_\_\_.

|  |
| --- |
| A．外部变量都是全局变量。 |
| B．静态的全局变量只能在本文件中使用，而不能在别的源文件中使用。 |
| C．使用外部变量之前，需要事先声明。 |
| D．通过extern关键字事先声明，可以使用其它源文件的静态变量。 |

1. 根据以下定义，表达式 w.h.d[5] 的值是\_\_\_\_\_\_\_\_\_。

struct {

int a, b, c[8];

struct { int d[6], f; } h;

} w = { 1, 2, {3, 4, 5}, { { 6, 7 }, 8 } };

|  |  |
| --- | --- |
| A．0 | B．2 |
| C．5 | D．8 |

1. 在单链表指针为p的结点之后插入指针为s的结点，正确的操作是\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．p->next=s; s->next=p->next; | B．s->next=p->next; p->next=s; |
| C．p->next=s; p->next=s->next; | D．p->next=s->next; p->next=s; |

1. 以下说法不正确的是\_\_\_\_\_\_\_\_\_\_\_。

|  |
| --- |
| A．头文件（.h文件）可以被单独编译。 |
| B．声明函数时，可以省略形式参数的名字，但是不可省略形式参数的类型。 |
| C．一个程序可以由多个源文件构成，其中.c文件通常包含一些全局变量的定义，以及一些函数的实现。 |
| D．头文件（.h文件）的内容通常包括宏定义、数据类型定义、全局变量声明和函数声明等。 |

1. 对于排序算法，下面说法正确的是\_\_\_\_\_\_\_.

|  |
| --- |
| A. 平均情况下，选择排序算法、冒泡排序算法的时间复杂度都是O( *N* 2)。 |
| B．平均情况下，选择排序算法、快速排序算法、的时间复杂度都是O( *N* log(*N*) )。 |
| C．平均情况下，归并排序算法、插入排序算法的时间复杂度都是O(*N*2 )。 |
| D．平均情况下，归并排序算法、冒泡排序算法的时间复杂度都是O( *N* log(*N*) )。 |

1. 在“文件包含”预处理语句的使用形式中，当#include后面的文件名用< >（尖括号）括起时，寻找被包含文件的方式是\_\_\_\_\_\_\_.

|  |
| --- |
| A．仅仅搜索当前文件夹。 |
| B．仅仅搜索源程序所在的文件夹。 |
| C．直接按系统设定的标准方式搜索文件夹。 |
| D．先在源程序所在文件夹搜索，再按系统设定的标准方式搜索。 |

1. 下列程序段的输出结果是\_\_\_\_\_\_\_.

#define A 10

#define B (A<A+2)-2

printf("%d", B\*2);

|  |  |  |  |
| --- | --- | --- | --- |
| A． 0 | B．2 | C．-2 | D．-3 |

试题二、问答题（共25分）

1. 下面的程序段定义了两个结构类型和一个指针变量，试图将第二个学生的生日赋值为1995，但不正确，请问错在何处，如何修正？（3分）

struct data {

int day, month, year;

};

struct student {

char name[20];

longnum;

struct data birthday;

};

struct student \*p = (struct student \*)malloc(10 \* sizeof(struct student));

\*(p+1)->birthday->year = 1995;

2. 下面的递归函数在实现逆序输出一个正整数的各位数字时有错误，请指出并修正。（3分）

void showdigits( unsigned int n )

{

while( n > 0 )

{

printf(“ %d ”, n%10);

showdigits(n/10);

}

}

3. 以下程序对字符串进行加密，加密算法是每一字符与15异或，实现低四位取反，高四位保持不变。请问是否正确，若不正确请说明原因，并修正。（4分）

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LENGTH 81

int main(void)

{

char \*p;

p = (char\*)malloc(sizeof(char)\*LENGTH);

strcpy(p, "I am a student!");

while (\*p++) \*p = \*p ^ 15;

free(p);

}

4. 下面是归并排序的核心模块：

void SortIntegerArray (int array[], int n)

{

inti, n1, n2, \*arr1, \*arr2;

if (n > 1) {

n1 = n / 2;

n2 = n – n1;

arr1 = NewArray (n1, int);

arr2 = NewArray (n2, int);

for (i = 0; i< n1; i++) arr1[i] = array[i];

for (i = 0; i< n2; i++) arr2[i] = array[n1 + i];

SortIntegerArray (arr1, n1);

SortIntegerArray (arr2, n2);

Merge (array, arr1, n1, arr2, n2);

FreeBlock (arr1);

FreeBlock (arr2);

}

}

对 (25, 50, 15, 35, 80, 85, 20, 40, 36, 70)进行归并排序，分别写出前4次调用Merge函数得到的合并数据情况。（4分）

5. 试说明数组与链表的区别，以及各自有什么优缺点。（6分）

6. 在C程序的多文件组织方式下，请问如何防止头文件被多重包含，具体做法是？（5分）

试题三、程序填空题（每空2分，共30分）

1. 已知有理数结构类型定义为：

typedef struct

{

int numerator; // 分子

int denominator; // 分母，不能是 0

} RATIONAL;

则有理数加法函数是（不考虑化简）：

RATIONAL AddRational(RATIONAL a, RATIONAL b)

{

RATIONAL t;

**(1)** ；

t.denominator = a.denominator \* b.denominator;

return t;

}

假设有理数化简函数是：RATIONAL SimplifyRational(RATIONAL a)，那么求n个有理数平均值的函数是（考虑化简）：

RATIONAL AverageRationals(RATIONAL a[], int n)

{

int i;

RATIONAL t = a[0];

for(i=1; i<n; i++)

**(2)** ；

t. numerator /= n;

**(3)** ；

return t;

}

2. 下面的程序根据读入的一个数字字符串，计算射击成绩。这样的数字字符串（由‘0’-‘9’组成）中的每个数字表示射击的环数成绩（约定最后两个数字不会是8和9）。用下面的计分规则计算成绩：8环以下只加入该环数，8环和9环除了加入该环数以外，还可以加上奖励分数：8环的奖励分是接下来两发射击的环数的较大者，9环的奖励分是接下来两发射击环数（较大者记为m，较小者记为n)的和值与差值的积（(m+n)\*(m-n)）。例如：程序若输入123，则输出grade = 6；输入1823，输出grade = 17；输入1923，输出grade = 20；输入18923，输出grade = 37。

#include <stdio.h>

#define MIN(a, b) a > b ? b : a

#define MAX(a, b) a > b ? a : b

#define MULT(x, y) 　 　 **(4)**  // 宏定义

#define CTOD(c) 　 　 **(5)**  // 宏定义

int main (void)

{

char s[80], \*p = s, c ;

int grade = 0, g, m, n;

gets(s);

while( c=\*p++ )

{

switch( g = CTOD(c) )

{

case 0: case 1: case 2: case 3: case 4: case 5: case 6: case 7:

grade += g; break;

case 8: m = MAX(CTOD (\*p), CTOD (\*(p+1)));

**(6)** ；;

break;

case 9: m = MAX(CTOD (\*p), CTOD (\*(p+1)) );

n = MIN(CTOD (\*p), CTOD (\*(p+1)) );

grade += g + MULT( m+n, m-n );

break;

}

}

printf("grade = %d\n", grade) ;

return 0;

}

3．在32bit图形模式下使用二维结构数组画点，以(300,200)为左上角，以(500,350)为右下角画一个实心的白色矩形。

#include <graphics.h>

typedef struct

{

char blue;

char green;

char red;

char zero;

} RGB;

void main()

{

int driver=0, mode=VESA\_1024x768x　 　 **(7)** bit;

int x, y;

RGB c = {0xFF, 0xFF, 0xFF, 0x00};

RGB (\*p)[1024];

initgraph(&driver, &mode, "");

p = (RGB (\*)[1024]) \_vp;

for(y=200; y<=350; y++)

{

for(x=300; x<=500; x++)

{

**(8)** ;

}

}

getchar();

**(9)** ;

}

4．小明编写了一个选择排序程序，一共有三个文件： sort.h，progmain.c, sort.c。为了方便查看程序是否正确运行，小明希望输出排序过程中的中间结果（即：在每一次选择操作之后，将数组的内容都打印出来）。一旦测试正确之后，他又希望能方便地关闭冗长的打印。因此，他利用了C语言的条件编译功能，打开或关闭输出。

这是一个多文件的程序，请完成空白处的语句，完成选择排序算法，使得程序能够正常编译链接，而且输出排序的中间结果。

文件 prog.c 的内容如下：

#include <stdio.h>

#include "sort.h"

int main()

{

int n, a[100], k;

printf("Input the number of data (<100): ");

scanf("%d", &n);

for( k=0; k<n; k++ )

scanf("%d", a+k);

selectsort(a, n);

return 0;

}

文件 sort.h 的内容如下：

**(10)**

void selectsort(int a[], int n);

void swap(int \*a, int \*b);

文件 sort.c 的内容如下：

**(11)**

void selectsort(int a[], int n)

{

int i, index, k

for(k=0; k<n-1; k++){

index=k;

for(i=k+1; i<n; i++)

if(a[i]< a[index]) index=i;

**(12)**  ;

EchoData(a,n);

}

}

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b; \*b = temp;

}

#ifdef \_ECHO

void EchoData(int a[], int n)

{

int k;

for( k=0; k<n; k++ ) printf("%d ", a[k]);

printf("\n");

}

#else

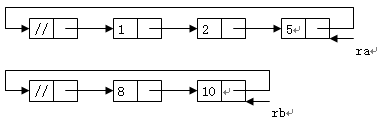
void EchoData(char s[])

{ }

#endif

5. 以下函数实现如图所示的功能：将用尾指针ra、rb表示的两个环形链表进行连接，返回连接后的环形链表的尾指针rb。

连接前：



连接后：



typedef int datatype;

typedef struct node

{

datatype data;

struct node \*next;

}linklist;

…...

linklist \*CONNECT(linklist \*ra, linklist \*rb)

{

**(13)**  ;

p=ra->next;

ra->next=　 　 **(14)**  ;

free(rb->next);

rb->next=　 　 **(15)**  ;

return(rb);

}

试题四、算法设计（15分）

1．冒泡法排序的改进版本是：当某轮次的比较结果不曾发生任何交换，就表明所有剩下的数据已经完全有序了，此时可以提前结束排序。请写出以下改进的冒泡排序函数BubbleSort2，函数BubbleSort2的原型如下：

void BubbleSort2(int a[ ], int n)； （5分）

2． 假设系统提供了3个功能函数：

1) 函数DrawTri，在屏幕上画一个三角形，声明如下：

void DrawTri( float u[2], float v[2], flow w[2], int color);

其中，参数u, v, w 表示三角形顶点的平面坐标；color表示画笔的颜色，color=0表示黑色，color=1表示白色。

2) 函数TriArea，计算三角形面积，声明如下：

float TriArea( float u[2], float v[2], flow w[2]);

其中，参数u, v, w 表示三角形顶点的平面坐标。

3)函数CalcMidPoint，计算线段中点，声明如下：

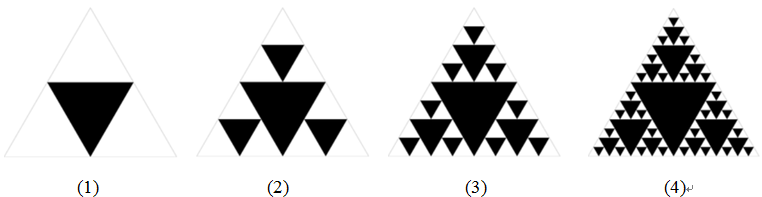
void CalcMidPoint(float a[2], float b[2], float m[2]);

其中，参数a, b 表示线段的两个端点平面坐标, 参数m保存中点的坐标。

请发现图形(1)~(4)的规律，设计绘制类似图形的算法。要求：1）简要描述算法思路；2）利用上述功能函数写出相应的实现函数Draw。函数Draw的原型如下：

void Draw( float u[2], float v[2], flow w[2], float t);

所绘制的图形中，最小的三角的面积不大于给定的阈值参数 t。 （10分）



注意：图(1)中黑色三角形顶点位于大三角形的中点。其它的以此类推。