**浙江大学2016–2017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： 2017 年 07 月 02 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3分，共30分）

1. 关于递归函数，下面的表述错误的是\_\_\_a\_\_\_\_.

|  |  |
| --- | --- |
| A．递归函数将直接调用自己(或间接) | B．递归函数必须有一个递归出口 |
| C．递归子问题与原问题形式相同 | D．系统实现递归需要额外内存开销 |

1. 某算法的时间复杂度为O(n2 )，表明该算法的\_\_\_c\_\_\_\_.

|  |  |
| --- | --- |
| A．问题规模是n2 | B．执行时间等于n2 |
| C．执行时间与n2 成正比 | D．问题规模与n2 成正比 |

1. 如下说法中正确的选择是\_\_\_c\_\_\_\_.

|  |
| --- |
| A. 可以在所有的变量前加上static，表明该变量是函数的局部变量 |
| B．修饰词static 与extern 在作为变量的修饰词时功能正好相反， 前者只能修饰局部变量，后者只能修饰全局变量 |
| C．修饰词static 与extern都可以修饰函数 |
| D．用static 修饰的变量是放在静态数据区的，而extern 修饰的变量是放在数据动态区的 |

1. 根据下面函数定义，调用f(3)得到的结果是\_\_d\_\_\_\_\_.

int f (int n)

{ return ( (n>0) ? f(n-1)+2\*f(n-2) : 1 ); }

|  |  |  |  |
| --- | --- | --- | --- |
| A．1 | B．5 | C．8 | D．11 |

1. 以下叙述中正确的是\_\_b\_\_\_\_\_.

|  |
| --- |
| A. 在程序的一行上可以出现多个有效的预处理命令行 |
| B．宏替换不占用运行时间，只占编译时间 |
| C．使用带参的宏时，参数的类型应与宏定义时的一致 |
| D．C语言的编译预处理就是对源程序进行初步的语法检查 |

1. 若有宏定义：#define M(x,y,z)  x/y-z ，则程序段的运行结果为\_\_\_b\_\_\_\_.

int a=2,b=3,c=5;

a=M(a\*a,b\*b,c\*c);

printf("%d\n",a);

|  |  |  |  |
| --- | --- | --- | --- |
| A．-21 | B．-22 | C．-24 | D．-25 |

1. 设有函数：void fun(int n,char \*s)；则下面对函数指针的定义和赋值均正确的是\_\_\_ad\_\_\_\_.

|  |  |
| --- | --- |
| A．void (\*pf)(int, char\*); pf=fun; | B．void \*pf(int, char\*); pf=fun; |
| C．void \*pf(int, char\*); \*pf=fun; | D．void (\*pf)(int,char\*); pf=&fun; |

1. 以下程序运行后，输出结果是\_\_d\_\_\_\_\_.

#include <stdio.h>

void main()

{

struct country

{ int num;

char name[10];

}x[5]={1,"China",2,"USA",3,"France",4, "England",5, "Spanish"},\*p;

p=x+2;

printf("%c",++p->name[2]);

}

|  |  |  |  |
| --- | --- | --- | --- |
| A．n | B．a | C．g | D．b |

1. 将指针p指向的节点插入到链表中指针q指向的节点后面，采用的语句是\_\_b\_\_\_\_\_.

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

1. 归并排序算法的时间复杂度是\_\_\_c\_\_\_\_.

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

试题二、改错题（每小题3分，共15分）

1. 请指出以下程序的错误，并修正。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #include <stdio.h>  #include <stdlib.h>  #define MAXLENGTH 80  int main()  {  /\*把存储在字符数组中的字符串小写字母变成大写字母\*/  char \*p = (char\*)malloc(MAXLENGTH+1);  while (\*p){  if (\*p>='a' && \*p<='z') \*p = \*p –('a'-'A');  p++;  }  printf("%s", p);  free(p);  return 0;  } |

1. 下面的递归函数欲实现一个数从高位开始逐位输出所有的位，如调用f(123)将输出123，请指出其中的错误并修正。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | void f(unsigned int n)  {  if (n<10) printf("%1d", n);  f(n/10);  printf("%1d", n%10);  } |

1. 假设一个工程文件中有文件a1.c和a2.c，编译连接运行后输出字符串“100World”. 请找出代码中的错误并修正。要求保留函数中的各条语句。

|  |  |  |  |
| --- | --- | --- | --- |
| 0  1  2  3  4  5  6  7  8  9 | /\*文件a1.c\*/  #include <stdio.h>  #include <string.h>  int g\_a;  int main()  {  char s1[80] ;  sprintf(s1,"%d",g\_a);  printf("%s", s1);  f2();  } | 0  1  2  3  4  5  6 | /\*文件a2.c\*/  #include <stdio.h>  int g\_a=100;  static void f2()  {  printf("World");  } |

1. 如下递归程序用来模拟完成hanoi塔的计算， 要求将64个盘从1号柱子移到2号柱子，请查看程序是否有错，有则改正（不要改变给出的printf的输出格式）。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | void hanoi (int n, char a, char b, char c)  { if (n == 1)  return;  else{  hanoi (n-1, a, c, b);  printf("%c-->%c\n", b,a);  hanoi (n-1, c, b, a);  }  }  int main(void)  {  hanoi (64, '1', ‘2', ‘3') ;  return 0;  } |

1. 请指出以下程序的错误，并修正。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | #include <stdio.h>  #define MAXLENGTH 81  int main()  {  int i;  char \*lines[10];  /\*输入10个有效长度<80的字符串\*/  for (i = 0; i<10; i++) {  gets(lines[i], MAXLENGTH);  }  return 0;  } |

试题三、问答题（共16分）

1. 为什么（1）中的结构类型定义是错误的，而（2）的结构类型定义是正确的？（2分）

|  |  |
| --- | --- |
| （1）struct A{  int num;  struct A b;  }; | （2）struct A{  int num;  struct A \*b;  }; |

1. 在交互式图形程序设计中，往往需要设计和实现哪四类回调函数？回调函数何时会被执行？（5分）
2. 如果采用二分查找算法在数组中查找元素，对数组有什么要求？该算法的最坏情况下时间复杂度是多少？给定有序序列（-5 -3 2 6 8 9 13 16 17 23 26 30 ）存于一数组中，用二分查找法查找关键字15的时候，需要比较的元素序列是怎样的？（6分）
3. 在C程序的多文件组织方式下，怎样通过预处理命令防止头文件被多重包含？（3分）

试题四、程序填空题（每空2分，共24分）

1. 以下函数SortLinkedList()采用选择排序方法，对链表按照从小到大进行排序。请根据题意完成程序填空。

typedef struct Node {

float value;

struct Node \* next;

} List;

void SortLinkedList(List \*head)

{

List \*h, \*p, \*pmin;

float temp;

for( h=head; h; h=h->next ) {

　 （1） 　 ;

for( p=h->next; p; p=p->next )

if( p->value < pmin->value )

　 （2） 　 ;

temp = pmin->value;

　 （3） 　 ;

h->value = temp;

}

}

1. 递归函数RSum可计算一个数组的前n个元素的和。在主程序中定义具有10个元素的数组b，要求调用RSum函数计算从b[1]开始的连续5个元素的和（即计算b[1]+…+b[5]的和）。请根据题意完成程序填空。

#include <stdio.h>

int RSum(int a[], int n)

{

if(　 （4） 　 )

return a[0];

else

return 　 （5） 　 ;

}

void main()

{

int s, b[10] = {-1,1,2,3,4,5,6,7,8,9};

s =　 （6） 　 ;

printf("%d\n",s);

}

1. 鸡尾酒排序，也就是双向冒泡排序，是冒泡排序的一种变形。此算法与冒泡排序的不同处在于排序时以双向在序列中进行排序。具体操作为：

数组中的元素本是无规律的存放，先把最小的元素定位到最前面，再把最大的元素定位到最后面；然后把第二小的定位，再定位次大元素。以此类推，直到完成排序。请根据题意完成程序填空。

void cocktail\_sort(int arr[], int len) {

int i, left = 0, right = len-1;

int temp;

while (　 （7） 　 ) {

for (i = left; i < right; i++)

if (　 （8） 　 ) {

temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

　 （9） 　 ;

for (i = right; i > left; i--)

if (arr[i - 1] > arr[i]) {

temp = arr[i];

arr[i] = arr[i - 1];

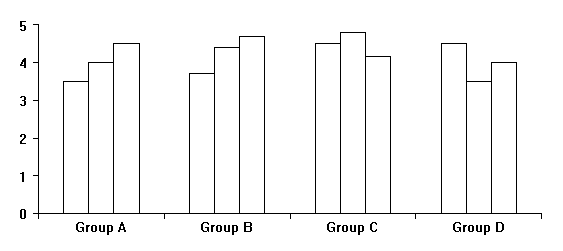
arr[i - 1] = temp;

}

left++;

}

}

1. 学校将2016级物理专业学生分为4个组：A、B、C、D，分别统计他们的数学、英语、计算机这3门课程的的平均成绩。李明同学编制下面的程序，将平均成绩绘制成“柱状图”。假设窗口坐标的y轴是向上的, 绘制的结果如下图所示。请根据题意完成程序填空。

#include "graphics.h"

#include <winuser.h>

/\*其他文件包含因篇幅所限未一一列出\*/

#define GroupNum 4 /\* 分组个数\*/

#define CourseNum 3 /\*课程个数\*/

#define MaxScore 5 /\*满分成绩\*/

float data[GroupNum][CourseNum] = { { 3.5, 4, 4.5 }, { 3.7, 4.4, 4.7 },

{ 4.5, 4.8, 4.2 },{ 4.5, 3.5, 4.0 } };

#define BAR\_WIDTH 20 /\*柱图的柱子宽度，20个像素\*/

#define UNIT\_HEIGHT 30 /\*1分成绩的高度，30个像素\*/

#define ScaleLen 5 /\* 刻度线的长度，5个像素\*/

#define SpaceX 15 /\*文字与坐标轴的水平间隔，15个像素\*/

#define SpaceY 15 /\*文字与坐标轴的垂直间隔，15个像素\*/

void DrawGraph(int x, int y);

void Main() /\*仅初始化执行一次\*/

{

SetWindowSize(800, 600);

　 （10） 　 ;

SetPenColor("Black");

SetPenSize(1);

DrawGraph(100,100);

}

/\*以（x, y）为坐标轴原点，绘制柱状图\*/

void DrawGraph(int x, int y)

{

int graph\_width = GroupNum \* (CourseNum+2) \* BAR\_WIDTH; /\* 总图宽度\*/

int graph\_height = MaxScore \* UNIT\_HEIGHT; /\*总图高度\*/

char str[16];

int g, k, strwidth;

/\* 垂直坐标轴 – y\*/

　 （11） 　 ;

DrawLine(0, graph\_height);

for (k = 0; k <= MaxScore; k++) {

int yk = y + k \* UNIT\_HEIGHT;

/\* 刻度线\*/

MovePen(x, yk);

DrawLine(- ScaleLen, 0);

/\*刻度值\*/

sprintf(str, "%d", k);

MovePen(x-SpaceX, yk-5);

DrawTextString(str);

}

/\*水平坐标轴 - x\*/

MovePen(x, y);

DrawLine(graph\_width, 0);

for (k = 0; k <= GroupNum; k++) {

int xk = x + k \* (CourseNum + 2) \* BAR\_WIDTH;

MovePen(xk, y);

DrawLine(0, - ScaleLen);

}

/\* 柱状图 - bars\*/

for (g = 0; g < GroupNum; g++) {

int xg = x + g \* (CourseNum + 2) \* BAR\_WIDTH;

for (k = 0; k < CourseNum; k++) {

DrawBox(xg + (k + 1) \* BAR\_WIDTH, y, BAR\_WIDTH,

(int)(data[g][k] \* UNIT\_HEIGHT));

}

/\*组名称\*/

sprintf(str, "Group %c", "ABCD"[g]);

MovePen(xg + ((CourseNum + 2) \* BAR\_WIDTH - TextStringWidth(str))/2,

y - SpaceY);

DrawTextString(str);

}

}

void DrawBox(int x, int y, int width, int height)

{

MovePen(x, y);

DrawLine(0, height);

DrawLine(width, 0);

　 （12） 　 ;

DrawLine(-width, 0);

}

试题五、算法设计（共15分）

1. 按照下面提供的部分代码，编写合并函数merge()，将两个升序排序的数组合并为一个升序数组。其中a1和n1表示第1个数组的指针和长度，a2和n2表示第2个数组的指针和长度。合并后的数组保存在数组m中。要求只能使用已经给出的局部变量，不能定义和使用新的局部变量。

void merge(int a1[], int n1, int a2[], int n2, int m[])

{

int p = 0, k1 = 0, k2 = 0;

/\* 以下代码部分需要自己完成 \*/

}

（1）描述函数merge()的实现思路。（2）完成该函数的定义。（5分）

2．假如有理数类型定义如下：

typedef struct

{

int n; // numerator,分子，允许是负数

int d; // denominator，分母，大于0

} RATIONAL;

现在需要编程实现两个有理数的加法功能，且要求计算结果是最简有理数形式（即分子与分母没有大于1的公因子）。若该功能相应的函数原型是：

RATIONAL AddR( RATIONAL a, RATIONAL b );

（1）描述函数AddR( )的实现思路。如果有必要则继续进行功能分解，增加其他的函数，与函数AddR( )构成结构化的上下级调用结构。（2）完成函数AddR( )及其下级函数（若有）的定义，并加入足量注释。（10分）