**一、选择题（每题1分，共40分）**

1.下列叙述中正确的是

A) 栈与队列都只能顺序存储 B) 循环队列是队列的顺序存储结构

C) 循环链表是循环队列的链式存储结构 D) 栈是顺序存储结构而队列是链式存储结构

【解析】栈是所有的插入与删除都限定在表的同一端进行的线性表；队列是指允许在一端进行插入，而在另一端进行删除的线性表，二者均即可以顺序存储也可以链式存储。为了充分地利用数组的存储空间，把数组的前端和后端连接起来，形成一个环形的表，称为循环队列，因此循环队列是队列的一种顺序存储结构， B选项正确。

2. 设某二叉树的前序序列为ABC，中序序列为CBA，则该二叉树的后序序列为

A) BCA B) CBA C) ABC D) CAB

【解析】二叉树遍历可以分为3种：前序遍历（访问根节点在访问左子树和访问右子树之前）、中序遍历（访问根节点在访问左子树和访问右子树两者之间）、后序遍历（访问根节点在访问左子树和访问右子树之后）。前序序列为ABC，则A为根节点。中序序列为CBA，则C和B均为左子树节点或右子树节点，且C为B父节点，可知后序序列为CBA，B选项正确。

3. 下列排序方法中，最坏情况下时间复杂度最小的是

A) 冒泡排序 B) 快速排序 C) 堆排序 D) 直接插入排序

【解析】在最坏情况下，对长度为n的线性表排序，冒泡排序、快速排序、直接插入排序的时间复杂度均为O(n^2)，堆排序时间复杂度为O(nlog2n)，复杂度最小。C选项正确。

4. 为了对有序表进行对分查找，则要求有序表

A) 只能顺序存储 B) 只能链式存储 C) 可以顺序存储也可以链式存储 D) 任何存储方式

【解析】二分法查找也称拆半查找，是一种高效的查找方法。能使用二分法查找的线性表必须满足两个条件：用顺序存储结构；线性表是有序表。A选项正确。

5. 软件设计中模块划分应遵循的准则是

A) 低耦合低内聚 B) 高耦合高内聚 C) 低耦合高内聚 D) 内聚与耦合无关

【解析】模块的独立程度可以由两个定性标准度量：内聚性和耦合性。耦合衡量不同模块彼此间互相依赖（连接）的紧密程度。内聚衡量一个模块内部各个元素彼此结合的紧密程度。一般来说，要求模块之间的耦合尽可能弱，即模块尽可能独立，且要求模块的内聚程度尽可能高。内聚性和耦合性是一个问题的两个方面，耦合性程度弱的模块，其内聚程度一定高。C选项正确。

6. 下面属于整数类的实例是

A) 0x518 B) 0.518 C) "-518" D) 518E-2

【解析】整数类实例包括：十进制常量用0~9表示，不能以0开头；八进制常量用0~7表示，必须用0开头；十六进制常量用0～9和A～F(a～f)表示，必须以0x或0X开头。A选项正确。B选项是浮点数实例，C选项是字符串实例，D选项为科学计数法表示的浮点数实例。

7. 下面属于白盒测试方法的是

A) 边界值分析法 B) 基本路径测试 C) 等价类划分法 D) 错误推测法

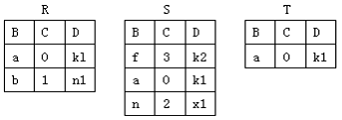
【解析】白盒测试是把程序看成装在一只透明的白盒子里，测试者完全了解程序的结构和处理过程。它根据程序的内部逻辑来设计测试用例，检查程序中的逻辑通路是否都按预定的要求正确地工作。白盒测试的主要技术有逻辑覆盖测试、基本路径测试等，B选项正确。常用的黑盒测试方法和技术有：等价类划分法、边界值分析法、错误推测法和因果图等，A、C、D选项错误。

8. 设有一个商店的数据库，记录客户及其购物情况，由三个关系组成：商品 (商品号，商品名，单价，商品类别，供应商) ，客户 (客户号，姓名，地址，电邮，性别，身份证号) ，购买 (客户号，商品号，购买数量)，则关系购买的键为

A) 客户号 B) 商品号 C) 客户号, 商品号 D) 客户号, 商品号, 购买数量

【解析】候选键（码）是二维表中能唯一标识元组的最小属性集。一个二维表有多个候选码，则选定其中一个作为主键（码）供用户使用。在购买关系中，商品号、客户号、购买数量均不能唯一标识元组，每一个单独的属性集都不能作为键，但商品号与客户号构成的集合与三者构成的集合都可以唯一标识购买关系，而前者为最小属性集，故商品号与客户号构成的集合为候选键，C选项正确。

9. 有两个关系R，S如下：



由关系R和S通过运算得到关系T，则所使用的操作为

A) 并 B) 自然连接 C) 差 D) 交

【解析】常用的扩充运算有交、除、连接及自然连接等。并：RS是将S中的记录追加到R后面。差：R-S结果是属于R但不属于S的记录组成的集合。交：RS结果是既属于R又属于S的记录组成的集合。自然连接：去掉重复属性的等值连接。交运算结果为a 0 k1，D选项正确。

10. 在数据库管理系统提供的数据语言中，负责数据的查询、增加、删除和修改等操作的是

A) 数据定义语言 B) 数据管理语言 C) 数据操纵语言 D) 数据控制语言

【解析】数据库管理系统提供了相应的数据语言，它们是：数据定义语言，该语言负责数据的模式定义与数据的物理存取构建；数据操纵语言，该语言负责数据的操纵，包括查询与增、删、改等操作；数据控制语言，该语言负责数据完整性、安全性的定义与检查以及并发控制、故障恢复等功能。C选项正确。

11. 流程图是描述算法的很好的工具，一般的流程图中由几种基本图形组成。其中判断框的图形是

A) 菱形 B) 长方形 C) 平行四边形 D) 椭圆型

【解析】算法可以用各种描述方法进行描述，目前最常用的有３种：伪代码、流程图和N-S结构图。传统的流程图下图所示的几种基本图形组成。A选项正确。

C:\Program Files\全国计算机等级考试上机考试题库 二级C语言\data\Cyuyan\Images\ms20-11-1.png

12. 以下叙述错误的是

A) 在进行模块化程序设计的时候，应首先完成每个模块的编写调试，再集中考虑主程序中的算法

B) 同一程序各模块可由一组人员同时进行编写调试，可提高编写程序的效率

C) 模块化的程序设计是采用自顶向下、逐步细化的原则

D) 程序的每个模块都可通过三种基本结构实现

【解析】结构化程序设计把一个复杂问题的求解过程分阶段进行，需要保证自顶向下、逐步细化、模块化设计、结构化编码。进行模块化设计时，首先设计框架，并定义和调试好各个模块之间的输入输出关系，然后完成各个模块的编写调试后再集中编译，模块化的程序设计采用自顶向下、逐步细化的原则，A选项叙述错误，C选项叙述正确。各个模块可以由不同人员同时进行编写调试，提高编写程序的效率，B选项叙述正确。结构化程序主要由3种基本控制结构组成，顺序结构、选择结构、循环结构，这三种基本结构可以解决任何复杂的问题，D选项叙述正确。故选择A选项。

13. 关于C语言的变量名，以下叙述正确的是

A) 变量名不可以与关键字同名 B) 变量名不可以与预定义标识符同名

C) 变量名必须以字母开头 D) 变量名是没有长度限制的

【解析】合法的标识符由字母（大、小写均可）、数字和下划线组成，并且必须以字母或下划线开头。关键字是指被Ｃ语言保留的，不能用作其他用途的标识符，它们在程序中都代表着固定的含义，用户不可重新定义，A选项正确。预定义标识符是系统已经有过定义的标识符，用户可以重新定义，所以可以作为变量名，B选项错误。变量名除了以字母开头以外，还可以以下划线开头，C选项错误。变量名没有长度限制，但不可超过编译器可以辨识的范围，D选项错误。

14. 有以下程序

#include <stdio.h> main()

{ int a;

a = (int)1.99\*3;

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

}

程序的运行结果是

A) 3 B) 5 C) 6 D) 结果不确定

【解析】浮点数1.99被强制转换成1，执行语句a=（int）1.99\*3；计算得a=1\*3=3，A选项正确。

15. 有以下程序

#include <stdio.h>

main()

{ int k=-17;

printf("%d,%o,%x\n", k, 1-k, 1-k);

}

程序的运行结果是

A) -17,22,12 B) -17,12,22 C) -17,-22,-12 D) 17,22,12

【解析】整型常量有3种表示方法，分别是十进制数表示法、八进制数表示法和十六进制数表示法。十进制整常量没有前缀，输出格式控制符为%d；八进制整常量以０作为前缀，输出格式控制符为%o；十六进制整常量以0X或0x作为前缀，输出格式控制符为%x。1-k=18，整型常量18用八进制表示为22，十六进制表示为12，A选项正确。

16. 若有定义：int a; float b; double c;，程序运行时输入：a=1,b=2,c=3 <回车>，能把值1输入给变量a、值2输入给变量b、值3输入给变量c的输入语句是

A) scanf("a=%d,b=%f,c=%lf", &a,&b,&c); B) scanf("%d%f%lf", &a,&b,&c);

C) scanf("a=%d,b=%lf,c=%lf", &a,&b,&c); D) scanf("a=%d,b=%f,c=%f", &a,&b,&c);

【解析】格式字符d，输入十进制整数；格式字符f，输入浮点数；格式字符lf，输入双精度浮点数。除了格式说明字符和附加格式字符外，如果还有其他字符，则在输入数据时要求按一一对应的位置原样输入这些字符。程序运行时输入：a=1,b=2,c=3，则scanf函数格式控制串对应的应该是“a=%d,b=%f,c=%lf”，A选项正确。

17. 有以下程序

#include <stdio.h>

main()

{ int a=0,b=0,c=0,d=0;

(++a || ++b) ? ++c : ++d;

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

}

程序的运行结果是

A) 1,0,1,0 B) 1,1,0,1 C) 1,0,0,1 D) 1,1,1,0

【解析】条件表达式一般形式：exp1?exp2:exp3，执行规则为：若exp1为真，求解exp2，输出exp2的值，不计算exp3的值；若为假，求解exp3，输出exp3的值，不计算exp2的值。本题程序执行过程为：求解逻辑表达式(++a || ++b)，先求解++a，a=1，则整个表达式为真，不计算++b，然后求解++c，不计算++d，最后得到a=1,b=0,c=1,d=0，A选项正确。

18. 有以下程序

#include <stdio.h>

main()

{ int a,b;

for (a=0; a<3; a++)

{ scanf("%d", &b);

switch(b)

{ case 1: printf("%d,", b++);

case 2: printf("%d,", b++);

default: printf("%d,", b++);

}

}

printf("\n");

}

执行时输入：1 2 3 <回车>，则输出结果是

A) 1,2,3,2,3,3, B) 1,2,3, C) 2,3,4,3,4,4, D) 1,1,1,2,2,3

【解析】本题执行过程为，输入1，匹配case1，输出1，然后执行b++得b=2,再执行下一个case的语句输出2，然后执行b++得b=3，继续执行default，输出3；输入2，执行case2，输出2，然后执行b++得b=3，执行default输出3；输入3，执行default输出3。A选项正确。

19. 有以下程序

#include <stdio.h>

main()

{ int a=1,b=1;

while(a--)

b--;

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

}

程序的运行结果是

A) -1,0 B) 0,0 C) -1,-1 D) 1,1

【解析】本题执行过程为：先取a值为1，执行循环，a=0，执行b--，b=0，再取a值为0，判断条件不成立，退出循环，再执行a--得到a=-1，最后输出a=-1,b=0。A选项正确。

20. 有以下程序

#include <stdio.h>

main()

{ int a=1, b=0;

for(;a<5; a++)

{ if (a%2==0) break;

b += a;

}

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

}

程序的运行结果是

A) 1 B) 10 C) 0 D) 4

【解析】程序的执行过程为：a=1时,a%2=1条件不成立，执行语句b+=a；得到b=1；a=2时,a%2=0条件成立，执行语句break；跳出循环。输出b=1，A选项正确。

21. 以下能正确输出字符a 的语句是

A) printf("%s", "a"); B) printf("%s", 'a'); C) printf("%c", "a"); D) printf("%d", 'a');

【解析】“格式控制串”用来指定每个输出项的输出格式，%s对应字符串，%c对应字符，%d对应整型。双引号里面的内容为字符串“a”，单引号里面的内容为字符'a'，A选项正确。B、C、D选项语句本身就有问题。

22. 有以下程序

#include <stdio.h>

main()

{ char c;

do {

c=getchar();

putchar(c);

}while(c != '#');

printf("\n");

}

执行时如输入：abcdefg##<回车> , 则输出结果是

A) abcdefg# B) abcdefg C) abcdefg## D) ##

【解析】程序执行过程为：读入一个字符存入缓存区，判断字符是否为“#”，如不是，继续读入字符，直到读入的字符是“#”,存入缓存区，退出循环，将缓冲区字符一个个输出，结果为abcdefg#，A选项正确。

23. 以下关于C语言函数参数传递方式的叙述正确的是

A) 数据只能从实参单向传递给形参 B) 数据可以在实参和形参之间双向传递

C) 数据只能从形参单向传递给实参

D) C语言的函数参数既可以从实参单向传递给形参，也可以在实参和形参之间双向传递，可视情况选择使用

【解析】数据只能由实参单向传递给形参称为“值传递”，而不能由形参传回来给实参，A选项正确；数组名、指针等作参数，实参传递给形参的是地址值，这样形参和实参就指向同一段内存单元，在函数体内对形参数据的改变也将影响到实参。实参和形参之间不可双向传递，B、D选项错误。数据不能从形参传递给实参，C选项错误。

24. 以下关于指针的叙述正确是

A) 所有类型的指针变量所占内存的大小是一样的

B) 指针变量所占内存的大小与其类型有关，char型指针变量只占1个字节，double型指针变量占8个字节

C) 指针变量可直接指向任何类型的变量，而不会出现编译或运行错误

D) 指针变量既可以直接指向结构体，也可直接指向结构体中某个成员，而不会出现编译或运行错误

【解析】专门用来存放变量地址的变量称为“指针变量”，所有地址值所占字节都是一样的，故指针变量所占内存大小一样，A选项正确，B选项错误。由于一个变量的地址（指针）还隐含有这个变量的类型信息，所以不能随意把一个地址存放到任何一个指针变量中去，只能把具有相同类型的变量的地址，存放到这个指针变量中，结构体变量与某个成员类型不同，不能用同样的指针指向它们，C、D选项错误。

25. 有以下程序

#include <stdio.h>

int k=5;

void f(int \*s)

{ s=&k; }

main()

{ int m=3,\*p=&m;

f(p);

printf("%d,%d\n", m, \*p);

}

程序的运行结果是

A) 3,3 B) 5,5 C) 3,5 D) 5,3

【解析】程序执行过程为：p指针指向变量m，调用函数，将m地址传给指针s，然后再让s指向变量k，而p指针所指并没有改变，输出为3,3，A选项正确。

26. 有以下程序

#include <stdio.h>

int \*f(int \*s,int \*t)

{ if (\*s < \*t) \*s=\*t;

return s;

}

main()

{ int i=3, j=5, \*p=&i, \*q=&j, \*r;

r=f(p,q);

printf("%d,%d,%d,%d,%d\n", i, j, \*p, \*q, \*r);

}

程序的运行结果是

A) 5,5,5,5,5 B) 3,5,5,5,5 C) 5,3,3,3,5 D) 3,5,3,5,5

【解析】程序执行过程为：p指向i，q指向j，调用函数f，将i与j地址传递给s与t，判断i是否小于j，成立则将j=5赋值给i，则i=5，返回指向i的指针并赋给r，则r指向i，最后输出5,5,5,5,5，A选项正确。

27. 有以下程序

#include <stdio.h>

#define N 4

void fun(int a[][N])

{ int b[N][N],i,j;

for(i=0; i<N; i++)

for(j=0; j<N; j++)

b[i][j] = a[N-1-j][i];

for(i=0; i<N; i++)

for(j=0; j<N; j++)

a[i][j] = b[i][j];

}

main()

{ int x[N][N]={ {1, 2, 3, 4},{5, 6, 7, 8}, {9,10,11,12},{13,14,15,16} },i;

fun(x); fun(x);

for (i=0;i<N; i++)

printf("%d,", x[i][i]);

printf("\n");

}

程序的运行结果是

A) 16,11,6,1, B) 1,6,11,16, C) 4,7,10,13, D) 13,10,7,4,

【解析】程序的执行过程为：调用函数f，将二维数组x地址传入函数，此函数实现将矩阵转置，然后将每一列首尾倒置，调用结果为x[N][N]={{13,9,5,1},{14,10,6,2},{15,11,7,3},{16,12,8,4}}。再次调用函数f，调用结果为x[N][N]={{16,15,14,13},{12,11,10,9},{8,7,6,5},{4,3,2,1}}。输出结果为16,11,6, 1，A选项正确。

28. 有以下程序

#include <stdio.h>

void fun(int a[ ], int n)

{ int i,j=0,k=n/2, b[10];

for (i=n/2-1; i>=0; i--)

{ b[i] =a[j]; b[k] =a[j+1]; j+=2; k++; }

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

a[i] = b[i];

}

main()

{ int c[10]={10,9,8,7,6,5,4,3,2,1},i ;

fun(c, 10);

for (i=0;i<10; i++)

printf("%d,", c[i]);

printf("\n");

}

程序的运行结果是

A) 2,4,6,8,10,9,7,5,3,1, B) 10,8,6,4,2,1,3,5,7,9, C) 1,2,3,4,5,6,7,8,9,10, D) 1,3,5,7,9,10,8,6,4,2,

【解析】程序的执行过程为：调用函数f，将数组c地址与n数值传入函数，此函数实现，将数组偶数位置元素从中间向前依次放入数组前半段，数组奇数位置元素从中间向后依次放入数组后半段，调用结果为c[10]={2,4,6,8,10,9,7,5,3,1},之后依次输出，A选项正确。

29. 有以下程序

#include <stdio.h>

#include <string.h>

main()

{ int i;

char a[]="How are you!";

for (i=0; a[i]; i++)

if (a[i] == ' ')

{ strcpy(a, &a[i+1]); i=0; }

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

}

程序的运行结果是

A) you! B) Howareyou! C) areyou! D) are you!

【解析】复制字符串函数strcpy(str1,str2)，将str2完整的（包括“\0”）复制到str1中，str1中原有的内容被覆盖。程序执行过程为：在a[i]不等于“\0”的情况下，判断当前元素是否为空格，若为空格将下一个元素以及之后的所有字符复制到a数组中。当i=3时，if条件成立，a以及之后单元中元素为“are you!\0”，i=0，进行下一次循环，之后当i=3时，if条件成立，a以及之后单元中元素为“you!\0”，i=0，之后的元素中在“\0”前没有空格出现，输出you!，A选项正确。

30. 有以下程序（其中的strstr()函数头部格式为：char \*strstr(char \*p1, char \*p2) 确定p2字符串是否在p1中出现，并返回p2第一次出现的字符串首地址）

#include <stdio.h>

#include <string.h>

char \*a="you";

char \*b="Welcome you to Beijing!";

main()

{ char \*p;

p = strstr(b, a) + strlen(a) + 1;

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

}

程序的运行结果是

A) to Beijing! B) you to Beijing! C) Welcome you to Beijing! D) Beijing!

【解析】调用strstr函数，可知a指向的字符串在b指向的字符串中第一次出现的位置，并将此地址赋给指针p。strlen()函数是求字符串的实际长度（不包含结束标志），并且作为函数值返回。strstr函数返回的地址下标值为9，加上a长度3，再加1，此时p指向的地址下标值为13，输出“to Beijing!”,A选项正确。

31. 有以下程序

#include <stdio.h>

main()

{ char b[4][10]; int i;

for (i=0; i<4; i++)

scanf("%s", b[i]);

printf("%s%s%s%s\n", b[0],b[1],b[2],b[3]);

}

执行时若输入: Fig flower is red. <回车>

则输出结果是

A) Figflowerisred. B) Figfloweris red. C) Figflower is red. D) Fig flower is red.

【解析】scanf()函数从输入设备按照指定的类型输入对应类型的若干个数据，遇到空格、制表符和回车时读取结束。可知读到第一个空格时b[0]赋值结束，为Fig，从第一个有效字符读到第二个空格时b[1]赋值结束，为flower，可知b[2]为is，b[3]为red.。printf()函数格式控制符%s输出字符串，遇到“\0”结束输出。输出结果为“Figflowerisred.”，A选项正确。

32. 有以下程序

#include <stdio.h>

#include <string.h>

main()

{ char w[20], a[5][10]={"abcdef", "ghijkl", "mnopq", "rstuv", "wxyz."};

int i;

for (i=0; i<5; i++)

w[i]=a[i][ strlen(a[i]) - 1];

w[5]= '\0';

puts(w);

}

程序的运行结果是

A) flqv. B) agmrw C) ekpuy D) flqvz

【解析】puts()函数将数组中存放的字符串输出到输出设备中，用“\n”取代字符串结束符“\0”。strlen()函数是求字符串的实际长度（不包含结束标志），并且作为函数值返回。程序中for循环实现，将第i个字符串最后一个字符放入w数组下标值为i的位置，即a[i][5]。最后将数组最后一个字符赋值为“\0”，输出字符串w为flqv，A选项正确。

33. 有以下程序

#include <stdio.h>

void f(int x[], int n)

{ if ( n> 1)

{ f(&x[1], n-1);

printf("%d,", x[0]);

}

else

printf("%d,", x[0]);

}

main()

{ int z[6] = {1,2,3,4,5,6};

f(z,6);

printf("\n");

}

程序的运行结果是

A) 6,5,4,3,2,1, B) 1,1 C) 1,1,1,1,1,1, D) 1,2,3,4,5,6,

【解析】在调用一个函数的过程中又出现直接或间接的调用该函数本身，称为函数的递归调用。本题目中递归函数调用执行过程为：f(z,6) ->f(&z[1],5) ->f(&z[2],4) ->f(&z[3],3) ->f(&z[4],2) ->f(&z[5],1)，输出x[0]，也即z[5]=6，返回执行f(&z[4],2)之后的输出x[0]语句，即5，同理再返回上一个f函数，依次输出4,3,2,1，A选项正确。

34. 有以下程序

#include <stdio.h>

int a=2;

int f()

{ static int n;

int m;

m=n=0;

n++; a++; m++;

return n+m+a;

}

main()

{ int k;

for (k=0; k<3; k++) printf("%d,", f( ));

printf("\n");

}

程序的运行结果是

A) 5,6,7, B) 5,7,9, C) 5,8,11, D) 5,5,5,

【解析】static变量又称为静态变量，编译时，将其分配在内存的静态存储区中，在整个程序运行期间都不释放这些存储单元，即使退出函数，下次再进入该函数时，静态局部变量仍使用原来的存储单元，值是上一次函数调用结束时的值。若定义时未赋初值，在编译时，系统自动赋初值为０；若定义时赋初值，则仅在编译时赋初值一次，程序运行后不再给变量赋初值。程序执行过程为：k=0,a=2,n=0,m=0;n=1,a=3,m=1,返回5；k=1,a=3,n=0, m=0;n=1,a=4,m=1,返回6；k=2,a=4,n=0,m=0;n=1,a=5,m=1,返回7；k=3退出循环。A选项正确。

35. 有以下程序

#include <stdio.h>

#define S(x) x\*x

main()

{ int k=5, j=2;

printf("%d,%d\n", S(k+j+2),S(j+k+2));

}

程序的运行结果是

A) 21,18 B) 81,81 C) 21,21 D) 18,18

【解析】带参数的宏的调用格式：宏名（实参表），替换过程是，用宏调用提供的实参字符串，直接置换宏定义命令行中相应形参字符串，非形参字符保持不变。S(k+j+2)被置换成k+j+2\*k+j+2，计算时先计算2\*k，结果为21；S(j+k+2)被置换成j+k+2\*j+k+2，计算时先计算2\*j，结果为18。程序的运行结果是21,18，A选项正确。

36. 有以下程序

#include <stdio.h>

typedef struct stu {

char name[10];

char gender;

int score;

} STU;

void f(STU \*a, STU \*b)

{ \*b = \*a;

printf("%s,%c,%d,", b->name, b->gender, b->score);

}

main()

{ STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};

f(&a,&b);

printf("%s,%c,%d\n", b.name, b.gender, b.score);

}

程序的运行结果是

A) Zhao,m,290,Zhao,m,290 B) Zhao,m,290,Qian,f,350

C) Qian,f,350,Qian,f,350 D) Qian,f,350,Zhao,m,290

【解析】程序执行过程为：调用f函数，将结构体a与b的地址作为参数传入函数，函数中指针a指向结构体a，指针b指向结构体b，将指针a指向的结构体a赋给指针b指向的结构体b，结构体b内容被修改。输出b指向的结构体b的元素Zhao, m,290。调用函数结束。输出结构体b的元素Zhao, m,290。A选项正确。

37. 有以下程序

#include <stdio.h>

#include <string.h>

typedef struct stu {

char name[10];

char gender;

int score;

} STU;

void f(char \*name, char gender, int score)

{ strcpy(name, "Qian");

gender = 'f';

score = 350;

}

main()

{ STU a={"Zhao", 'm', 290}, b;

b=a;

f(b.name,b.gender,b.score);

printf("%s,%c,%d,", a.name, a.gender, a.score);

printf("%s,%c,%d\n", b.name, b.gender, b.score);

}

程序的运行结果是

A) Zhao,m,290,Qian,m,290 B) Zhao,m,290,Zhao,m,290

C) Zhao,m,290,Qian,m,350 D) Zhao,m,290,Qian,f,350

【解析】程序执行过程为：将a赋给b，调用f函数，将b中name地址与gender和score的值传进函数，执行函数体，strcpy函数改变b中name值，但由于后两个成员为值传递不能改变b中gender和score值。调用函数结束后，结构体a={"Zhao", 'm', 290}，b={"Qian", 'm', 290}，A选项正确。

38. 若有定义：typedef int T[10]; T \*a[20]; 则与上述定义完全等价的说明语句是

A) int \*a[20][10]; B) int \*a[20]; C) int \*a[10]; D) int \*a[10][20];

【解析】新定义的类型T为大小为10的整型数组，定义T型指针数组\*a[20]，等价于定义了一个指向整型长度为10的数组的指针数组，这个指针数组大小为20，即为整型指针二维数组，行20列10，A选项正确。

39. 有以下程序

#include <stdio.h>

main()

{ int c,d;

c= 13&5; d=10&&5;

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

}

程序的运行结果是

A) 5,1 B) 1,1 C) 18,1 D) 13,1

【解析】“&”按位与运算，13=1101B，5=101B，执行语句c=13&5；结果为101B=5，即c=5。“&&”逻辑与运算，10与5均为真，结果为真，即d=1。A选项正确。

40. 有以下程序

#include <stdio.h>

main()

{ FILE \*fp;

int i, a[6]={1,2,3,4,5,6}, k;

fp = fopen("data.dat", "w+b");

fwrite(&a[0], sizeof(int), 1, fp);

for (i=1; i<6; i++)

{ fseek(fp, 0L, 0);

fread(&k, sizeof(int), 1, fp);

fseek(fp, 0L, 0);

a[i] +=k;

fwrite(&a[i], sizeof(int), 1, fp);

}

rewind(fp);

fread(&k, sizeof(int), 1, fp);

fclose(fp);

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

}

程序的运行结果是

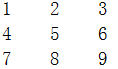
A) 21 B) 6 C) 123456 D) 11

【解析】程序执行过程为：以读/写方式打开一个新的二进制文件data.dat，从地址为a的数据块开始，一次输出一个整型字节的数据，只输出一次，将1写入文件中；执行for循环，将文件指针移到文件开头，将文件第一个数值读出赋给k，再将指针移动到开头，向文件内输入a[i]+k=3。for循环实现将a数组中元素累加，结果21存放在文件中。调用rewind将文件指针移动到开头，调用fread函数从文件中读出一个整型数据赋给k=21，调用fclose函数关闭文件，打印k值，A选项正确。

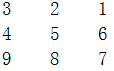
**二、程序填空题(18分)**

下列给定程序中，函数fun的功能是：将N×N矩阵主对角线元素的值与反向对角线对应位置上元素的值进行交换。

例如，若N＝3，有下列矩阵：



交换后为：



请在程序的下画线处填入正确的内容并将下画线删除，使程序得出正确的结果。

注意：部分源程序在文件BLANK1.C中。

不得增行或删行，也不得更改程序的结构！

#include <stdio.h>

#define N 4

/\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*/

void fun(int \_\_\_1\_\_\_ , int n)

{ int i,s;

/\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*/

for(\_\_\_2\_\_\_; i++)

{ s=t[i][i];

t[i][i]=t[i][n-i-1];

/\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*/

t[i][n-1-i]=\_\_\_3\_\_\_;

}

}

main()

{ int t[][N]={21,12,13,24,25,16,47,38,29,11,32,54,42,21,33,10}, i, j;

printf("\nThe original array:\n");

for(i=0; i<N; i++)

{ for(j=0; j<N; j++) printf("%d ",t[i][j]);

printf("\n");

}

fun(t,N);

printf("\nThe result is:\n");

for(i=0; i<N; i++)

{ for(j=0; j<N; j++) printf("%d ",t[i][j]);

printf("\n");

}

}

【参考答案】

(1)t[][N]　　(2)i=0;i<N　　(3)s

【考点分析】

本题考查：函数定义；for循环语句；数组元素的引用和赋值；变量值交换算法。

【解题思路】

填空1：根据main函数中的函数调用语句，可以填写fun函数定义中的形参内容。

填空2：由于是N×N矩阵，所以for语句中循环变量i的取值范围是0到N。

填空3：for循环体中是变量值交换算法。

**三、程序改错题(18分)**

由N个有序整数组成的数列已放在一维数组中，下列给定程序中函数fun的功能是：利用折半查找法查找整数m在数组中的位置。若找到，返回其下标值；否则，返回-1。

折半查找的基本算法是：每次查找前先确定数组中待查的范围low和high(low<high)，然后用m与中间位置(mid)上元素的值进行比较。如果m的值大于中间位置元素的值，则下一次的查找范围落在中间位置之后的元素中；反之，下一次的查找范围落在中间位置之前的元素中。直到low>high，查找结束。

请改正程序中的错误，使它能得出正确的结果。

注意：部分源程序在文件MODI1.C中，不得增行或删行，也不得更改的程序的结构！

#include <stdio.h>

#define N 10

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

void fun(int a[], int m )

{ int low=0,high=N-1,mid;

while(low<=high)

{ mid=(low+high)/2;

if(m<a[mid])

high=mid-1;

/\*\*\*\*\*\*\*\*\*\*\*\*found\*\*\*\*\*\*\*\*\*\*\*\*/

else If(m > a[mid])

low=mid+1;

else return(mid);

}

return(-1);

}

main()

{ int i,a[N]={-3,4,7,9,13,45,67,89,100,180 },k,m;

printf("a数组中的数据如下:");

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

printf("Enter m: "); scanf("%d",&m);

k=fun(a,m);

if(k>=0) printf("m=%d,index=%d\n",m,k);

else printf("Not be found!\n");

}

【参考答案】

(1)int fun(int a[],int m)或fun(int a[],int m)

(2)else if(m>a[mid])

【考点分析】

本题考查：折半查找算法；函数定义；if…else语句。

【解题思路】

(1)fun(int a[]，int m)函数的返回值为int类型，所以定义函数时，函数的返回类型不能是void，而是int类型。这里int可以省略，若省略函数类型标识符，系统将默认为int型。

(2)else If(m>a[mid])中，关键字if需要区别大小写，大写是错误的。

【解题宝典】

折半查找算法，将数列按有序化(递增或递减)排列，查找过程中采用跳跃式方式查找，即先以有序数列的中点位置为比较对象，如果要找的元素值小于该中点元素，则将待查序列缩小为左半部分，否则为右半部分。通过一次比较，将查找区间缩小一半。

**四、程序设计题(24分)**

假定输入的字符串中只包含字母和\*号。请编写函数fun，其功能是：除了尾部的\*号之外，将字符中的其它的\*号全部删除。形参p已指向字符串中最后的一个字母。在编写函数时，不得使用C语言提供的字符串函数。

例如，字符串中的内容为\*\*\*\*A\*BC\*DEF\*G\*\*\*\*\*\*\*，删除后，字符串中的内容应当是ABCDEFG\*\*\*\*\*\*\*。

注意：部分源程序在文件PROG1.C中。

请勿改动主函数main和其它函数中的任何内容，仅在函数fun的花括号中填入你编写的若干语句。

#include <stdio.h>

void fun( char \*a, char \*p )

{

}

main()

{ char s[81],\*t;

void NONO ( );

printf("Enter a string:\n");gets(s);

t=s;

while(\*t)t++;

t--;

while(\*t=='\*')t--;

fun( s , t );

printf("The string after deleted:\n");puts(s);

NONO();

}

void NONO()

{/\* 本函数用于打开文件，输入数据，调用函数，输出数据，关闭文件。 \*/

FILE \*in, \*out ;

int i ; char s[81],\*t ;

in = fopen("in.dat","r") ;

out = fopen("out.dat","w") ;

for(i = 0 ; i < 10 ; i++) {

fscanf(in, "%s", s) ;

t=s;

while(\*t)t++;

t--;

while(\*t=='\*')t--;

fun(s,t) ;

fprintf(out, "%s\n", s) ;

}

fclose(in) ;

fclose(out) ;

}

【参考答案】

void fun( char \*a, char \*p )

{

char \*t=a;

for(;t<=p;t++)

if(\*t!='\*')

\*(a++)=\*t;

for(;\*t!='\0';t++)

\*(a++)=\*t;

\*a='\0'; /\*在字符串最后加上字符串结束标识\*/

}

【考点分析】

本题考查：删除字符串中非尾部\*号，删除的主要思想就是把不删除的字符保留起来。

【解题思路】

本题用两个循环语句来实现。第1个循环的作用是将指针p所指字母以前所有非\*号的字符保留下来，即删除指针p以前所有的\*号。第2个循环的作用是将指针p以后的所有\*号保留下来。最后在新串的结尾加上结束符。