**答案**

## 一、单项选择题（共10题，40分）

1、C

解析：x后只能限1~2位十六进制数，用来表示ASCII码值为该十六进制数的字符，\x111的值为273，超出了ASCII码字符的允许范围。其他都是由三位以内（包括三位）的八进制数构成的转义字符。因此，本题答案选C。

2、D

解析：对于结构体a，它包含一个字符数组st[10]、一个整型int、一个双精度浮点型double和一个单精度浮点型float。根据这些信息，我们可以计算结构体a的总大小：char st[10]占用10个字节，int i占用4个字节，double z占用8个字节，float f占用4个字节。将这些加起来，结构体a的总大小为10 + 4 + 8 + 4 = 26字节。因此，本题答案选D。

3、C

解析：函数rewind用于将位置指针移到文件开头，其调用形式为rewind(fp)。因此，本题答案选C。

4、A

解析：使用括号将各部分进行划分，结果如下：（a<b)?a:（(c<d)?c:d）。第一个判断条件a<b成立，将执行第一个冒号左边的语句，该语句返回a的值，即1。因此，本题答案选A。

5、C

解析：程序设置变量x的初始值为-1，随后进入一个do-while循环，在循环中首先执行x=x\*x将x的值更新为1（因为-1乘以-1等于1），接着检查循环条件!x。由于x的值为1，!x计算结果为false，这意味着不满足循环条件，因此循环不会继续执行。所以，这段代码的循环只会执行一次。因此，本题答案选C。

6、B

解析：程序首先将变量f1和f2都初始化为1。接下来，进入一个循环四次的for循环，在循环体内，首先打印f1和f2的当前值，然后通过累加操作更新f1和f2的值，每次循环f1和f2的更新规则是f1得到f1+f2的结果，f2得到更新后的f1+f2的结果。按照这个逻辑，循环的四次输出分别是"1 1"、"2 3"、"5 8"和"13 21"，因此完整的输出结果是"1 1 2 3 5 8 13 21"，因此，本题答案选B。

7、B

解析：此题考察：数组—一维数组的定义及引用。定义数组时，[]中只能是常量表达式，A、C选项错误。数组定义时要确定数组的大小，D选项错误。因此，本题答案选B。

8、C

解析：代码中在调用函数前，通过int c(int x, int y)定义函数原型，在printf 函数中调用 c(a,b) 函数，函数需要传入两个参数,最终输出的是c函数中返回的结果。 c函数接受两个整型参数 a 和 b，在函数体内，使用了前置递减运算符（--）对 a 进行递减操作，然后将递减后的 a 和原始的 b 相加，并返回结果，因为 c(a,b) 的计算结果是 --a + b，即 0 + 2 = ，代码的输出结果将是2。因此，本题答案选C。

9、B

解析：使用scanf()函数给变量赋值时，需要提供变量的地址。选项A提供a的值，错误。选项B提供p的值，p作为指针变量其值为所指向变量的地址（即a的地址），正确。选项C提供p所指向变量的值（即a的值），错误。选项D提供指针变量p自身的地址，错误。因此，本题答案选B。

10、D

解析：指针运算符“\*”的优先级高于加法运算符“+”。“\*p+4”等价于“(\*p)+4”等价于a[0]+4，选项A错误，同理选项B错误。“\*(p+4)”代表p当前指向元素往后数的第4个元素，指针p指向数组第1个元素，往后第4个元素即数组的第5个元素，也是最后一个元素，选项C正确。“\*(p+5)”超出了数组的有效范围，选项D错误。因此，本题答案选C。

## 二、判断题 （共5题，15分）

11、正确

解析：C语言中，主函数的返回类型必须是int，表示程序的退出状态。因此，本题表述正确。

12、正确。

解析：逻辑与运算符短路机制：逻辑与运算符具有短路机制，即如果左侧表达式为假（结果为0），则不再评估右侧表达式或操作数，因为左侧结果已足以确定整个结果为假。因此，本题表述正确。

13、错误

解析：在C语言中，数组的下标必须是非负整数，不能使用负数作为下标。因此，本题表述错误。

14、正确

解析：在结构体中结构体变量所占内存空间的大小等于各成员变量所占空间之和。因此，本题表述正确。

15、正确

解析：在switch语句中，多个case可以共用一组执行语句，例如case 1:case 2:break;语句是合法的。因此，本题表述正确。

## 三、填空题 （共10题，20分）

16、#

解析：此题考察：程序设计和C语言—C语言程序结构。C语言的预处理语句以"#"开头。因此，划线处应填#。

17、7

解析：此题考察：数据表现形式及运算—字符型数据的定义和使用。在C语言中，字符串"china!"在内存中占用6个字节。这包括了6个字符（'c', 'h', 'i', 'n', 'a, '!'）以及一个用来表示字符串结尾的空字符('\0')。在C语言中，字符串以空字符'\0'结尾，因此需要一个额外的字节来存储这个空字符。因此，划线处应填7。

18、文本

解析：按照数据的组织形式，可以将数据文件分为文本文件和二进制文件。因此，划线处应填文本。

19、60

解析：首先解析表达式k=(++i)\*(j--)中的操作。++i是前缀递增，它会先将i的值加1，然后返回i的新值。所以 i 从5变成了6。j--是后缀递减，它会先返回 j 的当前值，然后将j减1。所以返回的值是10，之后j变成了9。现在我们可以计算表达式的值：k=(++i)\*(j--)=6\*10=60。因此，划线处应填60。

20、类型

解析：在定义使用结构体之前，我们需要先声明一个结构体类型，然后才能使用这个类型来定义变量。在结构体类型的声明中，需要指定了组成该结构体的数据字段及其类型。定义完成后，再根据声明的结构体类型来定义具体的结构体变量，这些变量将包含声明中指定的所有字段。因此，划线处应填类型。

21、a[1][0]

解析：此题考察：数组—二维数组的定义及引用。初始化的列表中只给出了每行的第一个元素，其余元素将被自动初始化为0。因此，数组a可以展开为：a[0][0]=2, a[0][1]=0, a[0][2]=0, a[1][0]=3, a[1][1]=0, a[1][2]=0。所以，数组元素a[1][0]的值为3。因此，划线处应填a[1][0]。

22、8921

解析：按照程序的逻辑，输入1298后，程序会进入一个循环，在这个循环中，程序通过取模运算得到n2的最低位数字并将其打印出来，随后通过整除运算去掉最低位。循环会持续进行，直到n2变为0。对于输入的1298，该循环会依次打印出8（1298%10的结果），然后是9（129%10的结果），接着是2（12%10的结果），最后是1（1%10的结果）。整个程序输出结果是8921。因此，划线处应填8921。

23、4 2 2 18 16

解析：此题考察：函数—数组作为函数参数的方法。示例代码中定义了一个整型数组arr，初始值设置为{2, 1, 1, 9, 8}。然后使用一个循环遍历数组中的每个元素，并将每个元素的值乘以2。最后调用了uo函数，uo函数接收两个参数：一个数组a和一个数组的长度s。在函数内部，使用另一个循环遍历数组中的每个元素，并使用printf函数打印出每个元素的值，以制表符分隔。由上述分析可知，整个程序的功能是将数组中的每个元素乘以2，并输出结果。因此，划线处应填4 2 2 18 16。

24、0 1 2 3 4 5 6 7 8 9

解析：此程序段定义了一个fun函数，使用了冒泡排序算法对数组a进行排序。冒泡排序会比较相邻的元素，并按照升序（从小到大）进行交换，这个过程会重复多次，直到没有可交换的元素为止。因此，给定的初始数组a中的元素通过fun函数被排序后，结果就是按照从小到大的顺序排列。当main函数调用fun之后，它会使用一个for循环来打印排序后的数组。数组中元素的顺序将是：0 1 2 3 4 5 6 7 8 9。因此，划线处应填0 1 2 3 4 5 6 7 8 9。

25、8

解析：函数func()中的while循环使用指针p对字符串中的每个字符进行访问，直到遇到空白字符。循环结束后，指针p将指向字符串结尾的空白字符，指针s指向字符串的首个字符。p-s得到两个指针之间的字符个数（即字符串长度），最后返回该值。字符串“abcd1234”的长度为8，因此，划线处应填8。

## 四、实验题 （共1题，15分）

26、①i<=9或i<10 ②num[i]<0 ③num[i]+sum或sum+num[i]

解析：此题考察：数组—一维数组的定义及引用。第一个空：需要遍历整个数组元素。由于数组下标为0-9共10个元素，所以for循环的遍历条件为：i<=9或i<10。第二个空：要求数组num中小于零的数据之和，那么在if语中要写条件num[i]<0来判断元素是否小于零。第三个空，将小于零的元素累加到sum中，所以填num[i]+sum;sum+num[i]。

## 五、应用题 （共1题，10分）

27、参考程序如下

#include <stdio.h>

int main() {

int i, j;

for(i = 1; i <= 9; i++) {

for(j = 1; j <= i; j++) {

printf("%d\*%d=%2d ", j, i, i \* j);

}

printf("\n");

}

return 0;

}

解析：程序使用两个嵌套的循环来实现。外层循环变量i从1到9，表示乘法表的每一行。内层循环变量j从1到i，表示每一行中的每一个乘法表达式。printf语句用于格式化输出，中间的%2d用于使单位和十位的数字能够对齐，使九九乘法表看起来整齐。