|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 关键字 | 2 | C语言程序的三种基本结构是（）。 | 顺序结构，选择结构，循环结构 | 递归结构，循环结构，转移结构 | 嵌套结构，递归结构，顺序结构 | 循环结构，转移结构，顺序结构 | A |
| 关键字 | 3 | 下列选项中，是C语言关键字的是（）。 | clude | while | scan | typee | B |
| 关键字 | 1 | C语言中提供的合法关键字是（）。 | swith | cher | kase | default | D |
| 执行流程 | 1 | C语言目标程序文件的扩展名是（）。 | c | obj | exe | lib | B |
| 执行流程 | 3 | 关于C程序的运行步骤正确的是（）。 | 可执行程序经过编译生成目标程序然后经过连接生成源程序 | 源程序经过编译生成目标程序然后经过连接生成可执行程序 | 源程序经过编译生成可执行程序然后经过连接生成目标程序 | 目标程序经过编译生成源程序然后经过连接生成可执行程序 | B |
| 执行流程 | 4 | C语言可执行文件程序文件的扩展名是（）。 | c | obj | exe | lib | C |
| 执行流程 | 5 | 一个C程序的执行是从（）。 | 本程序的main函数开始，到main函数结束 | 本程序的第一个函数开始，到最后一个函数结束 | 本程序的main函数开始，到最后一个函数结束 | 本程序的第一个函数开始，到main函数结束 | A |
| 执行流程 | 2 | C语言源程序文件的扩展名是（）。 | c | obj | exe | lib | A |
| 执行流程 | 6 | C语言可执行程序的开始执行点是（）。 | 程序中第一条可执行语句 | 程序中第一个函数 | 程序中的main函数 | 包含文件中的第一个函数 | C |
| 关于函数 | 9 | 以下叙述正确的是（）。 | 在C程序中，main函数必须位于程序的最前面 | C程序的每行中只能写一条语句 | C语言本身没有输入输出语句 | 在对一个C程序进行编译的过程中，可发现注释中的拼写错误 | C |
| 关于函数 | 10 | C语言程序的基本单位是（）。 | 程序行 | 语句 | 函数 | 字符 | C |
| 关于函数 | 6 | 一个完整的C源程序是（）。 | 要由一个主函数(或)一个以上的非主函数构成 | 由一个且仅由一个主函数和零个以上(含零)的非主函数构成 | 要由一个主函数和一个以上的非主函数构成 | 由一个且只有一个主函数(或)多个非主函数构成 | B |
| 关于函数 | 7 | 一条C语言的语句至少应包一个（）。 | {} | 逗号 | 分号 | 什么都不要 | C |
| 关于函数 | 8 | C语言规定：在一个源程序中，main函数的位置（）。 | 必须在最开始 | 必须在系统调用的库函数的后面 | 可以任意 | 必须在最后 | C |
| 关于函数 | 1 | 以下叙述正确的是（）。 | 在C程序中，main函数必须位于程序的最前面 | C程序的每行中只能写一条语句 | C语言本身没有输入输出语句 | 在对一个C程序进行编译的过程中，可发现注释中的拼写错误 | C |
| 关于函数 | 5 | 关于C程序，以下说法不正确的是（）。 | 每条语句后面必须有分号，变量声明后面不必加分号 | 可以用/\* \*/对C程序做注释 | C程序书写格式自由，一行内可以写多条语句 | C程序书写格式自由一条语句也可分多行书写 | A |
| 关于函数 | 10 | 一个C语言程序是由（）。 | 一个主程序和若干子程序组成 | 函数组成 | 若干过程组成 | 若干子程序组成 | B |
| 关于函数 | 3 | 关于C程序，以下说法不正确的是（）。 | 每条语句和数据声明后面必须有分号 | 注释语句是非执行语句 | C程序书写格式自由，一行内可以写多条语句 | 注释标志开始是\*/结束标志是/\* | D |
| 关于函数 | 4 | 关于C程序，以下说法不正确的是（）。 | 每条语句和数据声明后面必须有分号 | 可以用/\* \*/对C程序做注释 | C程序书写格式自由，一行内可以写多条语句 | C程序书写格式规定每行有且仅有一条语句 | D |
| 关于函数 | 2 | 以下叙述不正确的是（）。 | 一个C源程序可由一个或多个函数组成 | 一个C源程序必须包含一个main函数 | C程序的基本组成单位是函数 | 在C程序中，注释只能位于一条语句的后面 | D |
| 数据存储 | 5 | 字符(char)型数据在微机内存中的存储形式是（）。 | 反码 | 补码 | EBCDIC码 | ASCII码 | D |
| 数据存储 | 2 | 以下叙述不正确的是（）。 | 在内存中，数值是以它的补码存放的 | 负数的补码和该数的绝对值的补码相同 | 负数的补码是该数的绝对值的二进制数取反加1 | 正数的补码和它的原码相同 | B |
| 数据存储 | 1 | 以下叙述不正确的是（）。 | 在内存中，数值是以它的补码存放的 | 负数的补码和该数的绝对值的原码相同 | 负数的补码是该数的绝对值的二进制数取反加1 | 正数的补码和它的原码相同 | B |
| 数据存储 | 3 | 以下叙述不正确的是（）。 | 在内存中，数值是以它的补码存放的 | 正数的补码和它的原码相同 | 负数的补码是该数的绝对值的二进制数取反加1 | 在内存中，数值是以它的ASCII码存放的 | D |
| 数据存储 | 4 | 以下叙述不正确的是（）。 | 在内存中，数值是以它的补码存放的 | 正数的补码和它的原码相同 | 负数的补码是该数的绝对值的二进制数取反加1 | 负数的补码和它的原码相同 | D |
| 变量定义 | 7 | 设a=5，下列表达式结果为真的是 | a|a==a | a||a==a | a^a==a | ~a==a | A |
| 变量定义 | 14 | 设a=5，下列表达式结果为真的是 | a&&a==a | a||a==a | a^a==a | ~~a==a | D |
| 变量定义 | 9 | 设a=5，下列表达式结果为真的是 | a&&a==a | a||a==a | a^a^a==a | ~a==a | C |
| 变量定义 | 5 | 设a=5，下列表达式结果为真的是 | a&&a==a | a&a==a | a^a==a | ~a==a | B |
| 变量定义 | 2 | char a=7;<br>printf("%d\n",a&2);<br>上面程序段的输出结果为 | 1 | 2 | 4 | 7 | B |
| 变量定义 | 11 | char a=7;<br>printf("%d\n",a&5);<br>上面程序段的输出结果为 | 5 | 7 | 0 | 1 | A |
| 变量定义 | 12 | char a=7;<br>printf("%d\n",a&8);<br>上面程序段的输出结果为 | 0 | 1 | 2 | 4 | A |
| 变量定义 | 13 | char a=7;<br>printf("%d\n",a&4);<br>上面程序段的输出结果为 | 1 | 2 | 4 | 7 | C |
| 变量定义 | 10 | 关于变量，正确的说法是（）。 | 变量名是大小写区分的 | 变量可以不声明直接使用，也就是所谓的隐式声明 | 变量定义时可以不指定类型，系统会为它指定 | 变量与符号常量用法相同 | A |
| 变量定义 | 8 | 以下变量定义不正确的是（）。 | short c,d; | short c; int d=5; | short c=5;d=5; | short c,d=5; | C |
| 变量定义 | 15 | 以下变量定义正确的是（）。 | unsigned short c,d; | unsigned short c;d; | unsigned short c,unsigned short d; | unsigned short, c;unsigned short,d; | A |
| 变量定义 | 6 | 下列选项中，不是C语言关键字的选项是（）。 | IF | char | case | while | A |
| 变量定义 | 1 | C语言中的标识符只能由字母、数字和下划线三种字符组成，且第一个字符（）。 | 必须为字母 | 必须为下划线 | 必须为字母或下划线 | 可以是字母、数字或下划线 | C |
| 变量定义 | 3 | 以下变量定义不正确的是（）。 | short c,d; | short c; int d=5; | short c=5,d=5; | short c=d=5; | D |
| 变量定义 | 4 | 以下变量定义不正确的是（）。 | short c,d; | short c, int d=5; | short c=5,d=5; | short c,d=5; | B |
| 常量 | 8 | 以下变量定义不正确的是（）。 | short c:d; | short c; int d=5; | short c=5,d=5; | short c,d=5; | A |
| 常量定义 | 6 | 以下变量定义不正确的是（）。 | unsigned short c,d; | unsigned short c,d=5; | unsigned short c=5,d; | unsigned short c=d=5; | D |
| 常量 | 3 | 关于变量，不正确的说法是（）。 | 变量名是大小写区分的 | 变量在定义时要指明类型，以便为其安排合适的存储空间、进行合适的操作 | 变量定义时可以不指定类型，系统会为它指定一个默认类型 | 要求对所有用到的变量作强制定义，也就是"先定义，后使用" | C |
| 常量 | 7 | 在C语言程序中，可以用做变量名的是（）。 | 1e5 | a1 | int | \*p | B |
| 常量定义 | 1 | 关于变量，正确的说法是（）。 | 变量定义时可以不指定类型，系统会为它指定 | 变量可以不声明直接使用 | 变量名是大小写区分的 | 变量可以先使用后定义 | C |
| 常量 | 2 | 下面四个选项中，均是不合法的用户标识符的选项是（）。 | temp | \_A | b-a | INT | C |
| 常量 | 4 | 下列标识符中不合法的是（）。 | s\_name | \_e | fox | 3DS | D |
| 常量 | 5 | 关于变量，正确的说法是（）。 | 变量可以不声明直接使用，也就是所谓的隐式声明 | 变量可以被反复重新赋值 | 变量定义时可以不指定类型，系统会为它指定 | 变量可以先使用后定义 | B |
| 数据类型 | 4 | 若x、i、j和k都是int型变量，则计算下面表达式后，x的值为（）。<br>x=(i=4,j=16,k=32) | 4 | 16 | 32 | 54 | C |
| 数据类型 | 3 | 假设所有变量均为整型，则表达式（a=2,b=5,b++,a+b）的值是（）。 | 7 | 8 | 6 | 2 | B |
| 数据类型 | 5 | 设x、y、z和k都是int型变量,则执行表达式:x=(y=4,z=16,k=32)后,x的值为（）。 | 4 | 16 | 32 | 52 | C |
| 数据类型 | 2 | 若已定义x和y为double类型，则表达式:x=1，y=x+3/2的值是（）。 | 1 | 1.5 | 2.0 | 2.5 | C |
| 数据类型 | 6 | 下面程序段中，printf函数的输出结果为：（）<br>int x=10,y=3;printf("%d\n",y=x/y);} | 0 | 1 | 3 | 不确定 | C |
| 整型 | 1 | 设int x=1, y=1;表达式(!x||y--)的值是（）。 | 0 | 1 | 2 | -1 | B |
| 整型 | 5 | 设有 int x=11;则表达式(x++ \* 1/3)的值是（）。 | 3 | 4 | 11 | 12 | A |
| 整型 | 2 | 若有以下定义: char a; int b; float c; double d;<br>则表达式a\*b+d-c值的类型为（）。 | float | int | char | double | D |
| 整型 | 3 | 若变量a是int型变量，并执行了语句：a='A'+1;，下面叙述正确的是（）。 | a的值是字符型数据 | a的值是浮点型数据 | 不允许字符型和浮点型数据相加 | a的值是字符‘A’的ASCII值加1 | D |
| 整型 | 6 | 假设i为整型，f为单精度型，d为双精度型，e为长整型，<br>则表达式10+’a’+i\*f-d/e的值的数据类型是（）。 | int | float | double | 不确定 | C |
| 整型 | 7 | 若a=5,b=3,c=4,逻辑表达式!(a+b)&&!c的值为（）。 | 0 | -1 | 1 | 2 | A |
| 整型 | 4 | 表达式:10!=9的值是（）。 | true | 非零值 | 0 | 1 | D |
| 整型 | 9 | 若有代数式3ae/bc，则不正确的C语言表达式是（）。 | a/b/c\*e\*3 | 3\*a\*e/b/c | 3\*a\*e/b\*c | a\*e/c/b\*3 | C |
| 整型数 | 8 | 假设变量m为整型，n为单精度型，则以下能实现将n中的数值保留两位小数，对第三位进行四舍五入的表达式是（）。 | n=(n\*100+0.5)/100.0 | m=n\*100+0.5,n=m/100.0 | n(=n/100+0.5)\*100.0 | n=n\*100+0.5/100.0 | B |
| 赋值运算 | 1 | 请选出合法的C语言赋值语句（）。 | a=b=58 | i++; | a=58,b=58 | k=int a+b; | B |
| 赋值运算 | 3 | 下列可作为C语言赋值语句的是（）。 | x=3,y=5 | a=b=6 | i--; | y=int(x); | C |
| 赋值运算 | 4 | 以下符合C语言语法的赋值表达式是（）。 | d=9+e+f=d+9 | d=f=d+9 | d=9+e,e++,d+9 | d=9+e++=d+9 | B |
| 赋值运算 | 5 | 若有以下定义，则能使值为3的表达式是（）。<br>int k=7,x=12; | x%=(k%=5) | x%=(k-k%5) | x%=k-k%5 | (x%=k)-(k%=5) | D |
| 逗号运算 | 4 | 设x和y均为int型变量,则以下语句:x+=y;y=x-y;x-=y;的功能是（）。 | 把x和y按从大到小排列 | 把x和y按从小到大排列 | 无确定结果 | 交换x和y的值 | D |
| 逗号运算 | 2 | 设 a、b、c、d、m、n均为 int型变量，且 a=5、b=6、c=7、d=8、m=2、n=2<br>则逻辑表达式(m=a>b)&&(n=c>d)运算后，n的值为（）。 | 0 | 1 | 2 | 3 | C |
| 逗号运算 | 1 | 下列运算符其优先级最高的是（）。 | || | && | + | = | C |
| 混合运算 | 2 | 以下说法正确的是（）。 | 符号常量的定义位置为函数内部 | 符号常量一旦被定义将不能再被赋值 | 符号常量的定义属于C语句，应在结尾加分号 | 符号常量是个符号，可以被赋值 | B |
| 混合运算 | 6 | 以下说法不正确的是（）。 | 常量名的有效范围为定义命令之后到本源文件结束 | 符号常量的定义要用#define命令 | 符号常量的定义属于C语句，应在结尾加分号 | 符号常量一旦被定义将不能再被赋值 | C |
| 混合运算 | 1 | 下面数据不是合法常量的是（）。 | "printf" | "i" | "123.567" | true | D |
| 混合运算 | 4 | 以下说法正确的是（）。 | 符号常量的定义位置为函数内部 | 符号常量的定义位置为函数外部 | 符号常量的定义属于C语句，应在结尾加分号 | 符号常量是个符号，可以被赋值 | B |
| 混合运算 | 5 | 下面数据不是合法常量的是（）。 | "1e-5" | "i" | flase | 100.567 | C |
| 混合运算 | 3 | 下面数据不是合法常量的是（）。 | "printf" | "int" | true | "100.567" | C |
| 混合运算 | 7 | 以下说法不正确的是（）。 | 符号常量的定义位置为函数内部 | 符号常量的定义位置为函数外部 | 符号常量的定义不属于C语句，不需要在结尾加分号 | 符号常量名一般习惯用大写字母表示，以与变量名相区别 | A |
| 逻辑与关系运算 | 4 | 下面数据不是合法常量的是（）。 | "printf" | 'int' | 'a' | "100" | B |
| 逻辑与关系运算 | 2 | 下列选项不属于数值类型说明符的是（）。 | float | char | double | int | B |
| 表达式 | 1 | 下列选项不属于数值类型说明符的是（）。 | short | unsigned short | int | string | D |
| C语句 | 2 | 下列语句中不是合法的C语句的是（）。 | i++,j-- | {z=x+y;<br>t=z/100;<br>printf("%f",t); } | ; | Printf("C statement."); | A |
| C语句 | 1 | 下列语句中不是合法的C语句的是（）。 | for(i=0,i<=10,i++); | ; | i++; | a=b; | A |
| C语句 | 3 | 下列语句中不是合法的C语句的是（）。 | Printf("C statement."); | ; | a+b; | t=z/100 | D |
| C语句 | 4 | 下列语句中不是合法的C语句的是（）。 | i++; | Printf("C statement.") | {z=x+y;<br>t=z/100;<br>printf("%f",t); } | ; | B |
| printf函数 | 11 | printf函数中用到格式符%5s，其中数字5表示输出的字符串占用5列。<br>如果字符串长度大于5，则输出方式为（）。 | 从左边起输出该字符串，右补空格 | 按原字符长从左向右全部输出 | 右对齐输出该字符串，左补空格 | 输出错误信息 | B |
| printf函数 | 4 | 下面程序段中，printf函数的输出结果为：（）<br>int a=1234;<br>printf("%2d\n",a); | 12 | 34 | 1234 | 提示出错，无结果 | C |
| printf函数 | 8 | 执行下面程序段后printf函数的输出结果为（）。<br>int a=2,c=5;<br>printf("a=%d,b=%d\n",a,c); | a=%2,b=%5 | a=2,b=5 | a=d, b=d | a=%d,b=%d | B |
| printf函数 | 9 | printf("%.2f,%-8.2f \n",123.4567,123.4567);的输出结果为（）。 | 123.46,123.46□□□ | 123.46,123.46□□ | 123.46,□□□123.46 | 123.46,□□123.46 | B |
| printf函数 | 7 | printf("%-5.3s,%.4s\n", "CHINA","CHINA");的输出结果为（）。 | □□CHI ,CHIN | CHI□□,CHIN | CHINA,CHIN | CHINA,CHINA | B |
| printf函数 | 5 | printf("%7.2s,%.4s\n", "CHINA","CHINA");的输出结果为（）。 | □□□□□CH ,CHIN | CH□□□□□ ,CHIN | CH□□□□□ ,CHINA | □□□□□CH ,CHINA | A |
| printf函数 | 1 | 已有定义int a=-2;和输出语句printf("%8lx",a);<br>以下叙述正确的是（）。 | 整型变量的输出格式符只有%d一种 | %x是格式符的一种，它可以适用于任何一种类型的数据 | %x是格式符的一种，使数据按十六进制输出，但%8lx是错误的 | %8lx不是错误的格式符，其中8规定了输出字段的宽度 | D |
| printf函数 | 3 | 下面程序段中，printf函数的输出结果为：（）<br>char c='z';<br>printf("%c",c-25); | a | Z | z-25 | y | A |
| printf函数 | 6 | 以下程序的输出结果是（）<br>main()<br>{printf("\n\*s1=%15s\*","chinabeijing");<br>printf("\n\*s2=%-5s\*","chi");} | \*s1=chinabeijing□□□\*<br>\*s2=\*\*chi\* | \*s1=chinabeijing□□□\*<br>\*s2=chi□□\* | \*s1=chinabeijing□□□\*<br>\*s2=□□chi\* | \*s1=□□□chinabeijing\*<br>\*s2=chi□□\* | B |
| printf函数 | 2 | 语句:printf("%d",(a=2)&&(b=-2));的输出结果是（）。 | 无输出 | 结果不确定 | -1 | 1 | D |
| scanf函数 | 1 | 在scanf函数中，表示本输入项在读入后不赋给相应变量的说明字符是（）。 | L | h | c | \* | D |
| scanf函数 | 3 | x、y、z被定义为char型变量，若从键盘给x、y、z输入数据，正确的输入语句是 （）。 | input x,y,z; | scanf("%c%c%c",&x,&y,&z); | scanf("%d%d%d",x,y,z); | read("%d%d%d",&x,&y,&z); | B |
| scanf函数 | 4 | 关于scanf函数以下说法正确的是（）。 | 第二个参数可以是一个实型常量，如<br>scanf("%f",3.5); | 只有格式控制没有输入项，也能正确输入，如<br>scanf("%f"); | 当输入一个实型数据时，格式控制部分应指定实数的精度，如<br>scanf("%4.2f",&f); | 当输入数据时，必须指明变量的地址，如<br>scanf("%f",&f); | D |
| scanf函数 | 16 | 用下面scanf函数输入数据使x=1.5,y=-3.75,z=67.8,选项中输入正确的是（）。<br>  scanf("%f%f%\*f,%f",&x,&y,&z); | 1.5□-3.75□2.5,67.8 | 1.5□-3.75□2.5□67.8 | 1.5,-3.75,2.5,67.8 | 1.5□-3.75,67.8 | A |
| scanf函数 | 7 | 已有定义：float a,b;假设输入数据的方式为4.52（回车）3.5<br>下面输入语句正确的是（）。 | scanf("%f,%f",&a,&b); | scanf("%f%f",&a,&b); | scanf("%3.2f,%2.1f",&a,&b); | scanf("%3.2f%2.1f",&a,&b); | B |
| scanf函数 | 9 | 用下面scanf函数输入数据使x=10,y=20,c1='A',c2='B'，选项中输入正确的是（）。<br>  scanf("%d%d%c%c",&x,&y,&c1,&c2); | 1020AB | 10□20□A□B | 1020□AB | 10□20AB | D |
| scanf函数 | 5 | 以下scanf函数调用语句不合法的是（）。 | scanf("%7.2f %f ",&a,&b); | scanf("%ld %d ",&a,&b); | scanf("%3d %d ",&a,&b); | scanf("%3d %\*d %d",&a,&b); | A |
| scanf函数 | 8 | 在下面程序的横线处填入合适的语句使得程序运行结果如下：<br>输入A□B□C输出A□B<br>main( )<br>{char ch1,ch2,ch3;<br>\_\_\_\_\_\_\_\_\_\_\_<br>  printf("%c%c%c",ch1,ch2,ch3);  } | scanf("%c,%c,%c",&ch1,&ch2,&ch3); | scanf("%c□%c□%c",&ch1,&ch2,&ch3); | scanf("%c%c%c",&ch1,&ch2,&ch3); | scanf("%c%c",&ch1,&ch2,&ch3); | C |
| scanf函数 | 15 | 用下面scanf函数输入数据使x=1.5,y=-3.75,z=57.8,选项中输入正确的是（）。<br>  scanf("%f%f%\*f,%f",&x,&y,&z); | 1.5,-3.75,1.5,57.8 | 1.5□-3.75□1.5□57.8 | 1.5□-3.75□57.8 | 1.5□-3.75□1.5,57.8 | D |
| scanf函数 | 6 | 有输入语句scanf("a=%d,b=%d",&a,&b);<br>为使变量a,b的值分别为1和3<br>下面从键盘输入数据的形式正确的是（）。 | 13 | 1,3 | a=1□b=3 | a=1,b=3 | D |
| scanf函数 | 12 | 以下scanf函数调用语句合法的是（）。 | scanf("%3d %\*d %d",&a,&b); | scanf("%3d %d",a,b); | scanf("%3d %d ",\*a,\*b); | scanf("%3d %&d ",&a,&b); | A |
| scanf函数 | 11 | 用下面scanf函数输入数据使x=10,y=20,c1='A',c2='B'，选项中输入错误的是（）。<br>  scanf("%d%c%d%c",&x,&c1,&y,&c2); | 10A□20B | 10□A□20□B | 10A20B | 10A□□20B | B |
| getchar函数 | 1 | 下面说法正确的是（）。 | getchar函数可以得到一个返回值 | getchar函数可以完全接收从键盘上输入的数值15 | getchar函数可以接收用户输入的一串字符 | getchar函数是有参数的函数 | A |
| getchar函数 | 2 | getchar函数的功能是（）。 | 接收从键盘输入的字符 | 将从键盘输入的字符打印到屏幕上 | 接收从键盘输入的字符串 | 将从键盘输入的字符串打印到屏幕上 | A |
| putchar函数 | 1 | 关于putchar函数的说法错误的是（）。 | 该函数只能向屏幕上输出一个字符 | 该函数可以向屏幕上输出一个字符串 | 该函数是库函数 | 该函数可以向屏幕上输出控制字符 | B |
| putchar函数 | 2 | putchar函数的功能是（）。 | 接收从键盘输入的字符 | 将字符打印到屏幕上 | 将字符串打印到屏幕上 | 接收从键盘输入的字符串 | B |
| 关系逻辑运算 | 6 | 设a、b和c都是int型变量，且a=3，b=4，c=5，则下面的表达式中，值为0的表达式是  （）。 | 'a' && 'b' | a<=b | a || b+c && b-c | !((a<b) && !c || 1) | D |
| 关系逻辑运算 | 1 | 表示关系x<=y<=z的c语言表达式为（）。 | (x<=y)&&(y<=z) | (x<=y)AND(y<=z) | (x<=y<=z) | (x<=y)&(y<=z) | A |
| 关系逻辑运算 | 15 | 若要求在if后一对圆括号中表示a不等于0,则能正确表示这一关系的表达式为（）。 | a<>0 | !a | a=0 | a | D |
| 关系逻辑运算 | 16 | 能正确表示逻辑关系："a≥=10或a≤0"的C语言表达式是  （）。 | a>=10 or a<=0 | a>=0|a<=10 | a>=10 && a<=0 | a>=10 || a<=0 | D |
| 关系逻辑运算 | 19 | 设x、y、t均为int型变量，则执行语句：<br>x=y=3;t=++x||++y;后，y的值为（）。 | 不定值 | 4 | 3 | 1 | D |
| 关系逻辑运算 | 10 | 已有定义:int x=3,y=4,z=5;<br>则表达式!(x+y)+z-1 && y+z/2的值是  （）。 | 6 | 0 | 2 | 1 | D |
| 关系逻辑运算 | 7 | 能正确表示a和b同时为正或同时为负的关系或逻辑表达式是（）。 | (a>=0||b>=0)&&(a<0||b<0) | (a>=0 && b >=0) && (a<0 &&b <0) | (a+b>0) && (a+b<=0) | a\*b>0 | D |
| 关系逻辑运算 | 8 | 当c的值不为0时，在下列选项中能正确将c的值赋给变量a、b的是（）。 | c=b=a; | (a=c)||(b=c); | (a=c) && (b=c); | a=c=b; | C |
| 关系逻辑运算 | 3 | 在以下一组运算符中，优先级最高的运算符是（）。 | <= | = | % | && | C |
| 关系逻辑运算 | 12 | 设有定义：int a=2,b=3,c=4;<br>则以下选项中值为0的表达式是（）。 | (!a==1)&&(!b==0) | (a<b) && !c || 1 | a && b | a || (b+b) && (c-a) | A |
| 关系逻辑运算 | 14 | 当把以下四个表达式用作if语句的控制表达式时，有一个选项与其它三个选项含义不同，这个选项是（）。 | k%2 | k%2==0 | (k%2)!=0 | !(k%2)==0 | B |
| 关系逻辑运算 | 9 | 若x和y代表整型数，以下表达式中不能正确表示数学关系<br>|x-y|<10的是（）。 | abs(x-y)<10 | x-y>-10 && x-y<10 | (x-y)<-10||!(y-x)>10 | (x-y)\*(x-y)<100 | C |
| if语句 | 9 | 已有定义：float x=2.0,y;在执行下面的if语句后变量y的值为（）。<br>if(x<0.0) y=0.0;<br>else if(x<10.0) y=1.0/x;<br>else y=1.0; | 0.000000 | 0.250000 | 0.500000 | 1.000000 | C |
| if语句 | 7 | 下列条件语句中，功能与其他语句不同的是（）。 | if(a) printf("%d\n",x);<br>else printf("%d\n",y); | if(a==0) printf("%d\n",y);<br>else printf("%d\n",x); | if(a!=0) printf("%d\n",x);<br>else printf("%d\n",y); | if(a==0) printf("%d\n",x);<br>else printf("%d\n",y); | D |
| if语句 | 1 | 假定所有变量均已正确说明，下列程序段运行后x的值是（）。<br>a=b=c=0;x=35;<br>if(!a) x--;<br>else if(b);<br>if(c) x=3;else x=4; | 34 | 4 | 35 | 3 | B |
| if语句 | 3 | 在下列选项中，不正确的赋值语句是（）。 | ++t; | n1=(n2=(n3=0)); | k=i==j; | a=b+c=1; | D |
| if语句 | 5 | 若k是int型变量，且有下面的程序片段：<br>k=-3;<br> if(k<=0) printf("####")<br> else printf("&&&&");<br>上面程序片段的输出结果是（）。 | #### | &&&& | ####&&&& | 有语法错误，无输出结果 | D |
| if语句 | 8 | 已有定义：int a=2,b=-1,c=2;在执行下面的if语句后变量c的值为（）。<br>if(a<b)<br>if(b<0) c=0; else c++; | 0 | 1 | 2 | 3 | C |
| if语句 | 4 | 关于下面程序，说法正确的是（）。<br>main()<br>{int x=3,y=0,z=0;<br>if(x=y+z) printf("\* \* \* \*");<br>else printf("# # # #");} | 有语法错误不能通过编译 | 输出\* \* \* \* | 可以通过编译，但是不能通过连接，因而不能运行 | 输出# # # # | D |
| if语句 | 6 | 以下4个选项中，不能看作一条语句的是（）。 | {;} | a=0,b=0,c=0; | if(a>0); | if(b==0) m=1;n=2; | D |
| switch语句 | 3 | 运行以下程序后,如果从键盘上输入china#,则输出结果为（）。<br>main( )<br>{ int v1=0,v2=0; char ch; <br>while ((ch=getchar())!='#')<br> switch (ch )<br>{ case 'a':<br>case 'h':<br>default: v1++;<br>case '0':v2++; }<br> printf("%d,%d\n",v1,v2);} | 2,0 | 5,0 | 5,5 | 2,5 | C |
| switch语句 | 4 | 下列说法中正确的是（）。 | 在switch语句中一定要使用break语句 | 在switch语句中不一定要使用break语句 | break语句是switch语句的一部分 | break只能用于switch语句中 | B |
| switch语句 | 5 | 执行以下程序段后，变量a,b的值分别为（）。<br>int x=1,a=0,b=0;<br>switch(x)<br>{<br>case 0: b++;<br>case 1: a++;<br>case 2: a++;b++;<br>} | a=2,b=1 | a=1,b=1 | a=1,b=0 | a=2,b=2 | A |
| 条件运算符 | 4 | 以下程序段中printf的输出结果是（）。<br>int a=4,b=5,c=0,d;<br>d=!a&&!b||!c;<br>printf("%d\n",d); | 1 | 0 | 非0的数 | -1 | A |
| 条件运算符 | 2 | 以下程序段中printf的输出结果是（）。<br>int a=5,b=4,c=6,d;<br>printf("%d\n",d=a>b?(a>c?a:c):(b)); | 5 | 4 | 6 | 不确定 | C |
| 条件运算符 | 3 | 若给定条件表达式(M)?(a++):(a--)，则其中表达式M和（）等价。 | (M==0) | (M==1) | (M!=0) | (M!=1) | C |
| 读程序 | 2 | 以下程序的输出结果是（）。<br>main()<br>{int a=-1,b=4,k;<br>k=(++a<0)&&!(b--<=0);<br>printf("%d%d%d\n",k,a,b);} | 104 | 103 | 003 | 004 | D |
| 读程序 | 1 | 两次运行下面的程序,如果从键盘上输入6,则输出结果是（）。<br>main()<br>{int x;<br>scanf("%d",&x);<br>if(x++>5) printf("%d",x);<br>else printf("%d\n",x--);} | 7 | 6 | 4 | 3 | A |
| 读程序 | 3 | 若执行下面的程序时从键盘上输入5，则输出是（）。<br>main()<br>{int x;<br>scanf("%d",&x);<br>if(x++>5) printf("%d\n",x);<br>else printf("%d\n",x--);} | 7 | 6 | 5 | 4 | B |
| for较易 | 4 | 执行语句:for(i=l;i++<4;);后,变量i的值是（）。 | 3 | 4 | 5 | 不确定 | C |
| for较易 | 1 | 以下程序段执行后sum的值是（）。<br>int i,sum;<br>for(i=1;i<6;i++) sum+=i; | 15 | 14 | 不确定 | 0 | C |
| for较易 | 2 | 若i为整型变量，则以下循环执行次数是（）。<br>for(i=2;i==0;)printf("%d",i--); | 无限次 | 0次 | 1次 | 2次 | B |
| for较易 | 5 | 以下循环体的执行次数是（）。<br>main()<br>{ int i,j;<br>for(i=0,j=1; i<=j+1; i+=2, j--)<br>printf("%d \n",i); } | 3 | 2 | 1 | 0 | C |
| for较易 | 3 | 以下正确的描述是（）。 | continue语句的作用是结束整个循环的执行 | 只能在循环体内和switch语句体内使用break语句 | 在循环体内使用break语句或continue语句的作用相同 | 从多层循环嵌套中退出时，只能使用goto语句 | B |
| for较易 | 6 | 有如下程序<br>main()<br>{ int i,sum=0;<br>for(i=1;i<=3;sum++)<br>sum+=i;<br>printf("%d\n",sum);}<br>关于该程序，正确的说法是（）。 | sum的值为6 | i的值为4 | 该程序会陷入死循环 | sum的值为3 | C |
| do较易 | 4 | 假定a和b为int型变量,则执行以下语句后b的值为（）。<br>a=1;b=10;<br>  do<br>  {b-=a;a++;}<br>  while(b--<0); | 9 | -2 | -1 | 8 | D |
| do较易 | 7 | 在执行以下程序时，如果从键盘上输入：ABCdef(回车)，则输出为（）。<br># include<stdio.h> <br>main() <br>{ char ch;<br>while((ch=getchar())!='\n')<br>{if(ch>='A'&& ch<='Z'){ch=ch+32;printf("%c",ch);}}<br>} | ABCdef | abcdef | abc | DEF | C |
| do较易 | 7 | 在C语言程序中，当do-while语句构成的循环中的条件为（）时，结束循环。 | 0 | 1 | true | 非0 | A |
| do较易 | 6 | 执行下面程序片段的结果是（）。<br>int x=23;<br>do<br>{printf("%2d",x--);}<br> while(!x); | 打印出321 | 打印出23 | 不打印任何内容 | 陷入死循环 | B |
| do较易 | 1 | 下面程序的执行结果是（）。<br>main()<br>  {int x=23;<br>  do<br>  printf("%d",x--);<br>  while(!x);} | 321 | 23 | 不输出任何内容 | 陷入死循环 | B |
| do较易 | 5 | 以下叙述正确的是（）。 | do-while语句构成的循环不能用其它语句构成的循环来代替 | do-while语句构成的循环只能用break语句退出 | 用do-while语句构成的循环,在while后的表达式为非零时结束循环 | 用do-while语句构成的循环,在while后的表达式为零时结束循环 | D |
| do较难 | 2 | 执行以下程序后,输出结果是（）。<br>main()<br>  {int y=2;<br>  do<br>  {y--;}<br>  while(--y);<br>  printf("%d\n",y--);} | -1 | 1 | 2 | 0 | D |
| do较难 | 5 | 有以下程序段<br>int n=0,p;<br>do<br>{scanf("%d",&p);n++;}<br>while(p!=12345 && n<3);<br>此处do—while循环的结束条件是（ ）。 | p的值不等于12345并且n的值小于3 | p的值等于12345并且n的值大于等于3 | p的值不等于12345或者n的值小于3 | p的值等于12345或者n的值大于等于3 | D |
| do较难 | 1 | 执行以下程序段时（）。<br>x=-1;<br>do <br>{x=x\*x;}<br>while(!x); | 循环体将执行一次 | 循环体将执行两次 | 循环体将执行无限次 | 系统将提示有语法错误 | A |
| do较难 | 4 | 以下程序段的输出结果是（）。<br>int x=3;<br>do<br>{ printf("%3d",x-=2);}<br>while(!(--x)); | 1 | 3和0 | 1和-2 | 死循环 | C |
| do较难 | 6 | main()<br>{int s=0,a=1,n;<br>scanf("%d",&n);<br>do<br>{s+=1;a=a-2;}<br>while(a!=n);<br>printf("%d\n",s);  }<br>若要使程序的输出值为2，则应该从键盘给n输入的值是（ ）。 | -1 | -3 | -5 | 0 | B |
| while | 1 | 下列程序段的输出结果为（）。<br>int y=1;<br>while(y--);<br> printf("y=%d\n",y); | y=-1 | y=0 | 死循环 | y=9 | A |
| while | 4 | 下面程序的输出结果是（ ）。<br>main()<br>  {int n=4;<br>  while(n--)printf("%d ",--n);} | 2 0 | 3 1 | 3 2 1 | 2 1 0 | A |
| while | 7 | t为int类型，进人下面的循环之前，t的值为0，有如下程序片段：while(t=l) { …… }<br>以下叙述中正确的是（）。 | 循环控制表达式的值为0 | 循环控制表达式的值为1 | 循环控制表达式不合法 | 以上说法都不对 | B |
| while | 3 | 下面程序段的输出结果是（）。<br>main()<br>  {int n=9;<br>  while(n>6) {n--;printf("%d",n);}<br>} | 987 | 876 | 8765 | 9876 | B |
| while | 12 | 在C语言程序中，当while语句构成的循环中的条件为（）时，结束循环。 | 0 | 1 | true | 非0 | A |
| while | 8 | 以下程序段中，while循环的循环次数是（）。<br>main()<br>{int i=0;<br>while(i<10)<br>{if(i<1) continue;<br>if(i==5) break;<br>i++;}<br>……} | 1 | 10 | 6 | 死循环，不能确定次数 | D |
| while | 5 | 设有程序段<br>int k=10;<br>while(k=0) k=k-1;<br>下面描述正确的是（）。 | while循环执行10次 | 循环是无限循环 | 循环体语句一次也不执行 | 循环体语句执行一次 | C |
| while | 10 | 有以下程序段<br>int k=0;<br>while(k=1)<br>k++;<br>while循环执行的次数是（）。 | 无限次 | 有语法错，不能执行 | 一次也不执行 | 执行1次 | A |
| while | 2 | 语句while(!E);中的表达式!E等价于（）。 | E==0 | E!=1 | E!=0 | E==1 | A |
| 一维数组定义 | 1 | 在以下定义中，正确的说明语句为（）。 | int a[3][4]; | int a[3,4]; | int a[ ][ ]; | int a[ ][4]; | A |
| 一维数组定义 | 3 | C语言中数组下标的下限是（）。 | 1 | 0 | 视具体情况 | 无固定下限 | B |
| 一维数组定义 | 2 | 设有数组定义: char array [ ]="China"; 则数组 array所占的空间为（）。 | 4个字节 | 5个字节 | 6个字节 | 7个字节 | C |
| 一维数组的定义 | 4 | 以下对一维整型数组a的定义正确的是（）。 | int a(10); | int n=10,a[n]; | int n;<br>scanf("%d",&n);<br>int a[n]; | #define SIZE 10;<br>int a[SIZE]; | D |
| 一维数组的定义 | 7 | 以下定义语句正确的是（ ）。 | int n=5,a[n][n]; | int a[ ][3]={{1,2},{3,4},{5,6}}; | int a[ ][3]; | int a[ ][ ]={ {1,2},{3,4},{5,6}} | B |
| 一维数组的定义 | 6 | 要求定义具有8个int类型元素的一维数组，下面定义语句错误的是（ ）。 | # define N 8<br>int a[N]; | # define N 3<br>int a[2N+2]; | int a[ ]={0,1,2,3,4,5,6,7}; | int a[1+7]={0}; | B |
| 一维数组的定义 | 5 | 对数组a有如下定义int a[ ]={0,1,2,3,4};则数组a的下标的上限是（）。 | 0 | 4 | 5 | 不确定 | B |
| 二维数组的定义 | 3 | 以下对二维数组a初始化的语句正确的是（）。 | int a[ ][ ]={{1,2},{3,4},{0}}; | int a[3][ ]={{1,2},{3,4},{0}}; | int a[ ][3]={{1,2},{ },{0}}; | int a[ ][3]={{1,2},{3,4},{0}}; | D |
| 二维数组的定义 | 2 | 若有说明：int a[2][3]={0};下面叙述正确的是（）。 | 数组的每个元素都可得到初值0 | 此说明语句不正确 | 只有数组的第一个元素可以得到初值0 | 数组的每个元素都可得到初值，但值不确定 | A |
| 二维数组的定义 | 1 | 以下对二维数组a的说明正确的是（）。 | int a[3][ ]; | float a(3,4); | double a[1][4]; | float a(3)(4); | C |
| 一维数组引用 | 2 | 执行下面程序段后数组元素a[2]的值是（）。<br>int i, a[10];<br>for(i=9;i>=0;i--)a[i]=10-i; | 1 | 2 | 7 | 8 | D |
| 一维数组的引用 | 10 | 若有说明：int a[10];则对数组a的正确引用是（）。 | a[10] | a[3.5] | a(5) | a[10-10] | D |
| 一维数组的引用 | 8 | 关于数组的说法错误的是（）。 | 数组元素的下标是从0开始的 | 数组定义时方括号中不能包含变量 | 定义数组时方括号中的常量表达式表示数组包含的元素个数 | 数组元素的下标是从1开始的 | D |
| 一维数组的引用 | 7 | 若有如下定义语句：  int a[10]={1,2,3,4,5,6,7,8,9,10};<br>  则对数组正确的引用是（）。 | a[10] | a[a[3]5] | a[a[9]] | a[a[4]+4] | D |
| 一维数组的引用 | 9 | 若有以下说明：<br>int a[12]={1,2,3,4,5,6,7,8,9,10,11,12};char c='a',d,g;<br>则数值为4的表达式是（）。 | a[g-c] | a[4] | a['d'-'c'] | a['d'-c] | D |
| 一维数组的引用 | 6 | 若想使数组a中全部元素值都为0，下列写法错误的是（）。 | int a[10]={0, 0,0,0,0,0,0,0,0,0}; | int a[10]={0}; | int a[10]; | int a[10]={0,0}; | C |
| 字符数组的赋值 | 11 | 有以下定义：<br>  char x[]="abcdefg";<br>  char y[]={'a','b','c','d','e','f','g'};<br>则正确的叙述为（）。 | 数组x和数组y等价 | 数组x和数组y的长度相同 | 数组x的长度大于数组y的长度 | 数组x的长度小于数组y的长度 | C |
| 字符数组的赋值 | 6 | 以下选项中，不能正确赋值的是（）。 | char s1[10];s1="Ctest"; | char s2[]={'C', 't', 'e', 's', 't'}; | char s3[20]="Ctest"; | char \*s4="Ctest\n"; | A |
| 字符数组的赋值 | 10 | 有以下的定义初始化语句<br>char num[ ]="9876543210";<br>  数组num所占的存储空间的字节数是（ ）。 | 9 | 10 | 11 | 以上三项都不对 | C |
| 字符数组的赋值 | 2 | 下面对字符数组的描述错误的是（）。 | 字符数组可以存放字符串 | 字符数组中的字符串可以整体输入输出 | 可以通过赋值语句用=对字符数组整体赋值 | 不可以用关系运算符对字符数组中的字符串比较 | C |
| 字符数组的赋值 | 8 | 不能把字符串:Hello!赋给数组b的语句是（）。 | char b[10]={'H','e','l','l','o','!'}; | char b[10];b="Hello!"; | char b[10];strcpy(b,"Hello!"); | char b[10]="Hello!"; | B |
| 字符数组的赋值 | 1 | 已有定义char str1[15];若要用scanf函数给数组str1赋值，下面选项正确的是（）。 | scanf("%s",&str); | scanf("%s",str1); | scanf("%s",str1[i]); | scanf("%s",&str1[i]); | B |
| 字符数组的赋值 | 7 | 下列描述中不正确的是（）。 | 字符型数组中可以存放字符串 | 可以对字符型数组进行整体输入、输出 | 可以对整型数组进行整体输入、输出 | 不能在赋值语句中通过赋值运算符"="对字符型数组进行整体赋值 | C |
| 字符数组的赋值 | 3 | 下述对C语言字符数组的描术中错误的是（）。 | 字符数组可以存放字符串 | 字符数组中的字符串可以整体输入、输出 | 可以在赋值语句中通过赋值运算符"="对字符数组整体赋值 | 不可以用关系运算符对字符数组中的字符串进行比较 | C |
| 字符数组的赋值 | 5 | 已有如下定义：<br>char str1[10],str2[ ]={"China"};<br>要将字符数组str2的值赋给字符数组str1，正确的赋值方式为（）。 | str1=str2; | str1="China"; | str1[ ]={"China"}; | strcpy(str1,str2); | D |
| scanf函数 | 1 | 有两个字符数组c、d,c、d为数组名，则以下正确的输入语句是（）。 | gets(c,d); | scanf("%s%s",c,d); | scanf("%s%s",&c,&d); | gets("c"),gets("d"); | B |
| scanf函数 | 3 | 有两个字符数组a,b，以下正确的输入语句是（）。 | gets(a,b); | scanf("%s%s",a,b); | scanf("%s%s",&a,&b); | gets("a"),gets("b"); | B |
| printf输出字符串 | 5 | 下面程序段的执行结果是（）。<br>char a[7]="abcdef";<br>char b[4]="ABC";<br>strcpy(a,b);<br>printf("%c",a[5]); | 空格 | \0 | e | f | D |
| printf输出字符串 | 6 | 下面程序段的运行结果是（）。<br>char s[5]={'a','b','\0','c','\0'};<br>printf("%s",s); | 'a''b' | ab | ab□c | 'a''b'□'c' | B |
| printf输出字符串 | 4 | 若有定义和语句: char s[10];s="abcd";printf("%s\n",s); 则结果是（）。 | 输出abcd | 输出a | 输出abcd□□□□□□ | 编译不通过 | D |
| printf输出字符串 | 3 | 下面程序段在执行后，printf函数的输出结果是（）。<br>char c[ ][8]={"hello","world"};<br>printf("%s\n", c[1]); | hello | world | hello world | 程序段有错误 | B |
| printf输出字符串 | 2 | 已有定义char str1[15]="hello";<br>若要用printf函数把数组str1所有元素的值输出，下面选项正确的是（）。 | printf("%s",str1); | printf("%s",&str1); | printf("%s",str1[i]); | printf("%s",&str1[i]); | A |
| puts输出字符串 | 1 | 执行下面程序段后会输出（）。<br>char str[ ]={"china\nbeijing"};<br>puts(str); | china<br> beijing | china□beijing | china | beijing | A |
| puts函数 | 2 | 有两个字符数组a,b，以下正确的输出语句是（）。 | puts(a,b); | printf("%s%s",a[],b[]); | putchar(a,b); | puts(a);puts(b); | D |
| 数组初始化 | 4 | 以下能对一维数组a正确初始化的语句是（）。 | int a[10]=(0); | int a[ ]={0}; | int a[10]={ }; | int a[10]={0\*10}; | B |
| 数组初始化 | 3 | 以下对二维数组a初始化的语句错误的是（）。 | int a[2][3]={0}; | int a[ ][3]={{1,2},{3,4},{0}}; | int a[2][3]={{1,2},{5,6}}; | int a[2][3]={{1,2},{ },{3,4}}; | D |
| 数组初始化 | 2 | 对以下说明语句int a[10]={6,7,8,9,10};正确的理解是（）。 | 将五个值依次赋给a[1]至a[5] | 将五个值依次赋给a[0]至a[4] | 将五个值依次赋给a[6]至a[10] | 此语句不正确 | B |
| 数组初始化 | 1 | 对二维数组的初始化下列选项错误的是（）。 | char ch[2][10] = {{"Hello"}, {"World"}}; | char ch[2][10] = "Hello", "World"; | char ch[2][10] = {"Hello", "World"}; | char ch[ ][10] ={ "Hello","World"}; | B |
| 函数缺省类型 | 4 | C语言允许函数值类型缺省定义，此时该函数值隐含的类型是（）。 | float | int | long | double | B |
| 函数缺省类型 | 2 | 在C语言中，变量的默认存储类别是（）。 | auto | static | extern | 无存储类别 | A |
| 函数缺省类型 | 3 | 在C语言中，若未对函数类型显式说明，则函数的隐含类型是（）。 | void | double | int | char | C |
| 函数定义 | 9 | 以下所列的各函数首部中，正确的是（）。 | void play(var:integer,var b:integer) | void play(int a,b) | void play(int a,int b) | Sub play(a as integer,b as integer) | C |
| 函数定义 | 4 | 以下叙述中不正确的是（）。 | 在不同的函数中可以使用相同名字的变量 | 函数中的形式参数是局部变量 | 在一个函数内定义的变量只在本函数范围内有效 | 在一个函数内的复合语句中定义的变量在本函数范围内有效 | D |
| 函数定义 | 8 | 以下说法中正确的是（）。 | C语言程序总是从第一个的函数开始执行 | 在C语言程序中,要调用的函数必须在main函数中定义 | C语言程序总是从main函数开始执行 | C语言程序中的main函数必须放在程序的开始部分 | C |
| 函数定义 | 2 | 以下正确的函数定义形式是（）。 | double fun(int x,int y) | double fun(int x; int y) | double fun(int x, int y); | double fun(int x,y); | A |
| 函数调用 | 2 | 有如下函数调用语句  func(rec1,rec2+rec3,(rec4,rec5));<br>该函数调用语句中，含有的实参个数是（）。 | 3 | 4 | 5 | 有语法错误 | A |
| 函数调用 | 7 | 以下说法正确的是（）。 | 用户若需要调用标准库函数，调用前必须重新定义 | 用户可以重新定义标准库函数，若如此，该函数将失去原有含义 | 系统根本不允许用户重新定义标准库函数 | 用户若需要调用标准库函数，调用前不必使用预编译命令将该函数所在文件包括到用户源文件中 | B |
| 函数调用 | 14 | 有以下函数定义：  void fun(int n, double x) {……}<br>若以下选项中的变量都已正确定义并赋值，则对函数fun的正确调用语句是（）。 | fun(int y,double m); | k=fun(10,12.5); | fun(x,n); | void fun(n,x); | C |
| 函数调用 | 13 | 以下错误的描述是（）。 | 函数调用可以出现在执行语句中 | 函数调用可以出现在表达式中 | 函数调用可以作函数的实参 | 函数调用可以作函数的形参 | D |
| 函数调用 | 15 | 自定义函数f的定义如下：<br>void f(int v, int w)<br>{int t;<br>t=v;v=w;w=t;}<br>若主调函数中有如下定义和调用语句<br>int x=4,y=3,z=5;<br>f(x,y);f(y,z);<br>则在调用后变量x,y,z的值分别是（）。 | 3,4,5 | 5,3,4 | 5,4,3 | 4,3,5 | D |
| 函数调用 | 12 | C语言中函数能否嵌套调用和嵌套定义（）。 | 二者均不可以 | 前者可以，后者不可以 | 前者不可以，后者可以 | 二者均可以 | B |
| 函数调用 | 8 | 以下程序段有语法错误，关于错误原因正确的说法是（）。<br>void main()<br>{  int G=5,k;<br>void prt\_char();<br>……<br>k=prt\_char(G);<br>……} | 语句void prt\_char();有错，它是函数调用语句，不能用void说明 | 变量名不能用大写字母 | 函数说明和调用语句之间有矛盾 | 函数名不能使用下划线 | C |
| 函数调用 | 10 | 以下对C语言函数的有关描述中，正确的是（）。 | 调用函数时，只能把实参的值传送给形参，形参的值不能传送给实参 | 函数既可以嵌套定义又可以嵌套调用 | 函数必须有返回值，否则不能使用函数 | 函数既可以嵌套定义不可以嵌套调用 | A |
| 函数声明 | 7 | 在一个C程序中（）。 | main函数必须出现在所有函数之前 | main函数可以在任何地方出现 | main函数必须出现在所有函数之后 | main函数必须出现在固定位置 | B |
| 函数声明 | 6 | 若程序中定义了以下函数double myadd(double a,double B){return (a+B);}<br>该函数放在调用语句之后，则在调用之前应该对该函数进行说明，以下说明错误的是（）。 | double myadd(double a,B); | double myadd(double,double); | double myadd(double b,double A); | double myadd(double x,double y); | A |
| 函数声明 | 1 | 下面说法错误的是（）。 | 形式参数属于局部变量 | 不同函数中可以使用相同名字的变量 | 主函数中不能使用其他函数中定义的变量 | 主函数中可以使用其他函数中定义的变量，但其它函数不能使用主函数中定义的变量 | D |
| 函数参数传递 | 5 | 在调用函数时，以下描述中不正确的是（）。 | 调用函数时，实参可以是表达式 | 调用函数时，将为形参分配内存单元 | 调用函数时，实参必须有确定的值 | 调用函数时，实参只能是变量 | D |
| 函数参数传递 | 2 | 若用以为数组名作函数实参，以下说法正确的是（）。 | 必须在主调函数中说明此数组的大小 | 实参数组和形参数组的类型可以不匹配 | 在被调函数中必须指明形参数组的大小 | 实参数组名与形参数组名必须相同 | A |
| 函数参数传递 | 6 | C语言规定，简单变量做实参时，它和对应形参之间的数据传递方式为（）。 | 地址传递 | 单向值传递 | 由实参传给形参，再由形参传回给实参 | 由用户指定传递方式 | B |
| 函数参数传递 | 1 | C语言规定，简单变量做实参时，它和对应形参之间的数据传递方式为（）。 | 地址传递 | 单向值传递 | 由实参传给形参，再由形参传回给实参 | 由用户指定传递方式 | B |
| 函数参数传递 | 4 | 以下说法正确的是（）。 | 实参和与其对应的形参各自占用独立的存储单元 | 实参和形参共同占用一个存储单元 | 只有当实参和与其对应的形参同名时才共同占用一个存储单元 | 形参是虚拟的，不占用存储单元 | A |
| 函数参数传递 | 7 | 以下说法正确的是（）。 | 实参和与其对应的形参占用独立的存储单元 | 实参和形参共同占用一个存储单元 | 只有当实参和与其对应的形参同名时才共同占用一个存储单元 | 形参是虚拟的，不占用存储单元 | A |
| 函数返回值 | 1 | 以下函数值的类型是（）。<br>fun(float x)<br>{float y;<br>y=3\*x-4;<br>return y;} | int | 不确定 | void | float | A |
| 函数返回值 | 3 | C语言中函数返回值的类型是由( )决定。 | return语句中的表达式类型 | 主调函数的类型 | 调用函数时临时指定 | 函数定义时所指定的函数类型 | D |
| 函数返回值 | 4 | C语言中函数返回值的类型是由( )决定。 | return语句中的表达式类型 | 主调函数的类型 | 调用函数时临时指定 | 定义函数时所指定的函数类型 | D |
| 函数返回值 | 5 | 若已定义的函数有返回值，则以关于该函数调用的叙述中错误的是（）。 | 调用可以作为独立的语句存在 | 调用可以作为一个函数的实参 | 调用可以出现在表达式中 | 调用可以作为一个函数的形参 | D |
| 函数填空题 | 5 | 有如下程序段<br>int func(int a,int b)<br>{return(a+b);}<br> main(){ int x=2,y=5,z=8,r;<br>r=func(func(x,y),z);}<br>在执行该程序段后变量r的值是（）。 | 12 | 13 | 14 | 15 | D |
| 函数填空 | 1 | 以下程序的功能是求出变量a和b的较大值，则在横线处需填写的内容是（）。<br>main()<br>  {int a=1,b=2,c;<br>\_\_\_\_\_\_\_\_\_\_<br>printf("%d\n",c);}<br>max(int x,int y)<br>{int z;  z=(x>y)?x:y;<br>return(z);} | c=max(a,b); | c=max(int a,int b); | c=max(int x,int y); | c=max(x,y); | A |
| 函数填空题 | 6 | 下面函数的功能是判断一个数是否为素数，横线处要填入的内容是（）。<br>int prime(int number)<br>{int flag=1,n;<br>for(n=2;n<=sqrt(number);n++)<br>if\_\_\_\_\_\_\_\_\_\_<br>{flag=0; break;}<br>return(flag);} | (number%n==0); | (number%n==0) | (number%n=0); | (number%n=0) | B |
| 函数填空题 | 3 | 下面程序的功能是（）。<br>void main()<br>{<br>int fac(int n);<br>int i;<br>for(i=1;i<=5;i++)<br>printf("%d\n",fac(i));}<br>int fac(int n)<br>{<br>static int f=1;<br>f=f\*n;<br>return(f);} | 输出1到5的和 | 输出5的阶乘 | 输出1到5的阶乘 | 输出1到5 | C |
| 函数填空题 | 8 | 下面函数的功能是判断一个数是否为素数，横线处要填入的内容是（）。<br>int prime(int number)<br>{int flag=1,n;<br>for(n=2;n<=sqrt(number);n++)<br>if(number%n==0)<br>{\_\_\_\_\_\_\_\_\_\_<br>break;}<br>return(flag);} | flag=0; | flag=1; | flag==0; | flag==1; | A |
| 函数填空题 | 9 | 下面程序的功能是输出1到5的阶乘，横线处需填写的内容是（）。<br>void main()<br>{<br>int fac(int n);<br>int i;<br>for(i=1;i<=5;i++)<br>printf("%d\n",fac(i));}<br>int fac(int n)<br>{<br>\_\_\_\_\_\_\_\_\_\_<br>f=f\*n;<br>return(f);} | static int f=0; | static int f=1; | auto int f=1; | auto int f=0; | B |
| 数组与函数 | 5 | 若用数组名作为函数调用的实参，传递给形参的是（）。 | 数组的首地址 | 数组第一个元素的值 | 数组中全部元素的值 | 数组元素的个数 | A |
| 数组与函数 | 3 | C语言中数组名作为参数传递时，作为实际参数的数组名被处理为（）。 | 该数组的长度 | 该数组的元素个数 | 该数组中各元素的值 | 该数组的首地址 | D |
| 基本概念 | 4 | 已知int i,a[5],\*p;<br>下面那个语句是不合法的。 | p=&i ; | p=a; | p=&a; | p=&(i+1); | D |
| 基本概念 | 3 | 已知int a[5],\*p;<br>下面那个语句是合法的。 | p=a+3; | p=a[3]; | p=\*(a+3); | p=&(a+3); | A |
| 基本概念 | 1 | 已知int i,a[5],\*p;<br>下面那个语句是合法的。 | p=a; | p=\*a; | p=\*a[5]; | a=p; | A |
| 基本概念 | 5 | 若有以下定义，则对a数组元素的正确引用是<br> int a[5],\*p=a; | \*&a[5] | a+2 | \*(p+5) | \*(a+2) | C |
| 基本概念 | 2 | 已知int a[5],\*p;<br>下面那个语句是合法的。 | p=&a[1]; | p=a[1]; | p=\*a[1]; | p=\*(a+1); | A |
| 指针运算 | 1 | 已有程序段<br>char a[5]="1234",\*p,\*p1;<br>p=a;p1=p+2;<br>printf("%d\n",p1-p);<br>输出结果是 | 1 | 2 | 3 | 4 | B |
| 指针运算 | 2 | 已有程序段<br>int a[5]={1,2,3,4,5},\*p,\*p1;<br>p=a;p1=p+3;<br>printf("%d\n",p1-p);<br>输出结果是 | 1 | 2 | 3 | 4 | C |
| 指针运算 | 3 | 已有程序段<br>int a[5]={1,2,3,4,5},\*p,\*p1;<br>p=a;p1=&a[3];<br>printf("%d\n",a[p1-p]);<br>输出结果是 | 1 | 2 | 3 | 4 | D |
| 指针运算 | 4 | 已有程序段<br>char a[5]="1234",\*p;<br>p=a;a[3]='\0';<br>printf("%s\n",p);<br>输出结果是 | 1 | 12 | 123 | 1234 | C |
| 指针运算 | 5 | 已有程序段<br>char a[5]="1234",\*p;<br>p=a;p+=2;<br>printf("%s\n",p);<br>输出结果是 | 34 | 1234 | 3 | 234 | A |
| 指针运算 | 6 | 已有程序段<br>char a[5]="1234",\*p;<br>p=a;p+=1;a[3]='\0'<br>printf("%s\n",p);<br>输出结果是 | 12 | 23 | 34 | 234 | B |
| 指针变量作函数参数 | 4 | 有以下程序段<br>void main()<br>{<br>int a=1,b=2;<br>void ch(int \*a,int \*b);<br>ch(&a,&b);<br>printf("%d,%d\n",a,b);<br>void ch(int \*a,int \*b)<br>{(\*a)++;(\*b)++;}<br>该程序段的输出结果是 | 1,2 | 2,3 | 1,3 | 2,2 | B |
| 指针变量作函数参数 | 3 | 有以下程序段<br>void main()<br>{<br>int a=1,b=2;<br>void ch(int \*a,int \*b);<br>ch(&a,&b);<br>printf("%d,%d\n",a,b);<br>void ch(int \*a,int \*b)<br>{a++;(\*b)++;}<br>该程序段的输出结果是 | 1,2 | 2,3 | 1,3 | 2,2 | C |
| 指针变量作函数参数 | 2 | 有以下程序段<br>void main()<br>{<br>int a=1,b=2;<br>void ch(int \*a,int \*b);<br>ch(&a,&b);<br>printf("%d,%d\n",a,b);<br>void ch(int \*a,int \*b)<br>{(\*a)++;b++;}<br>该程序段的输出结果是 | 1，2 | 2，3 | 2，2 | 1，3 | C |
| 指针变量作函数参数 | 1 | 有以下程序段<br>void main()<br>{<br>int a=1,b=2;<br>void ch(int \*a,int \*b);<br>ch(&a,&b);<br>printf("%d,%d\n",a,b);<br>void ch(int \*a,int \*b)<br>{a++;b++;}<br>该程序段的输出结果是 | 1，2 | 1，3 | 2，2 | 2，3 | A |
| 字符与指针 | 5 | 下面程序段的运行结果是<br>char \*s="abcd";<br>s+=1;<br> printf("%d",\*s);<br>已知a的ASCII码值为97。 | 97 | 98 | 99 | 100 | B |
| 字符与指针 | 4 | 下面程序段的运行结果是<br>char \*s="abcd";<br> printf("%c",\*s); | a | b | c | d | A |
| 字符与指针 | 3 | 下面程序段的运行结果是<br>char \*s="abcd";<br>s+=1;<br> printf("%c",\*s); | a | b | c | d | B |
| 字符与指针 | 1 | 下面程序段的运行结果是<br>char \*s="abcde";<br>s+=2;<br> printf("%s",s); | cde | 字符'c' | 字符'c'的地址 | 无确定的输出结果 | A |
| 字符与指针 | 2 | 下面程序段的运行结果是<br>char \*x="cdefg";<br>x+=2;<br> printf("%s",x); | efg | 字符'e' | 字符'e'的地址 | EFG |  |
| 数组与指针 | 5 | 已有程序段：<br>int a[2][2]={1,2,3,4},\*p,i,s;<br>  p=&a[0][0];<br>for(i=0;i<2;i++)<br>{s=s+\*p;p++;}<br>s的结果为 | 二维数组a所有元素的和 | 二维数组a的第一行元素的和 | 二维数组a的第一列元素的和 | 二维数组a的第二行元素的和 | B |
| 数组与指针 | 4 | 已有程序段：<br>int a[2][2]={1,2,3,4},\*p,i,s;<br>  p=&a[0][0];<br>for(i=0;i<4;i++)<br>{s=s+\*p;p++;}<br>s的结果为 | 二维数组a所有元素的和 | 二维数组a的第一行元素的和 | 二维数组a的第一列元素的和 | 二维数组a的第二行元素的和 | A |
| 数组与指针 | 3 | 以下语句中存在语法错误的是 | char ss[10]; ss[10]= "hello"; | char ss[10]={"hello"}; | char \*ss; ss= "hello"; | char \*ss= {"hello"}; | A |
| 数组与指针 | 2 | 以下语句中存在语法错误的是 | char ss[6][20]; ss[1]= "hello"; | char ss[ ][20]={"hello"}; | char \*ss[6]; ss[1]= "hello"; | char \*ss= {"hello"}; | A |
| 数组与指针 | 1 | 以下语句中存在语法错误的是 | char ss[6][20]; ss[1]= "right?"; | char ss[ ][20]={"right?"}; | char \*ss[6]; ss[1]= "right?"; | char \*ss= {"right?"}; | A |
| 基本概念 | 2 | 下面选项中，均是正确的八进制数或十六进制数的选项是（）。 | -10<br>0x8f <br>011 | 0abc <br>017 <br>0xc | 010<br>0x11 <br>0xf1 | 0a12<br>0x123<br>0xa | C |
| 基本概念 | 3 | 在C语言中，不正确的int类型的常数是（）。 | 32768 | 0 | 037 | 0xaf | A |
| 基本概念 | 4 | 以下数据属于整型常量的是（）。 | "0x12" | '0x12' | 0x12 | '\x12' | C |
| 基本概念 | 1 | 以下数据属于整型常量的是（）。 | "012" | '012' | '\012' | 012 | D |
| 基本运算 | 1 | 以下数据不属于整型常量的是（）。 | 0123 | 123 | 0x123 | x123 | D |
| 基本运算 | 2 | 设C语言中，一个int型数据在内存中占2个字节，则unsigned int型数据的取值范围是（）。 | 0～255 | 0～32767 | 0～65535 | 0～2147483647 | C |
| 基本运算 | 4 | 设C语言中，一个int型数据在内存中占2个字节，则int型数据的取值范围是（）。 | -2147483648～2147483647 | 0～32767 | 0～65535 | -32768～32767 | D |
| 基本运算 | 3 | 下面四个选项中，不合法的浮点数是（）。 | 0.12 | 2e4.2 | 123e4 | 0.0 | B |