

C语言程序设计

单选(521)--(80)

- 1、char 类型的长度为 ()。>[A.1](#)
- 2、char 类型的长度为 () 个字节。>[A.1](#)
- 3、C 程序中的哪些是预处理指令： ()。>[A. #include, #define, #endif](#)
- 4、C 语言中使用的字符常量，其起止标记符是 ()。>[A.单引号](#)
- 5、C 语言 if 语句中的表达式是： ()。>[A. 可以是任意表达式](#)
- 6、C 语言程序由函数组成，它的 ()。>[B. 主函数可以在其它函数之后，函数内不可以嵌套定义函数。](#)
- 7、C 语言程序中的基本功能模块为 ()。>[A.函数](#)
- 8、C 语言的流程控制语句不包括 ()。>[A.表达式语句](#)
- 9、C 语言源程序的基本单位是 ()。>[B. 函数](#)
- 10、C 语言源程序文件的扩展名为 ()。>[D.c](#)
- 11、C 语言源程序文件的缺省扩展名为 ()。>[D. .c](#)
- 12、C 语言中 while 和 do-while 循环的主要区别是：()。>[D. do-while 的循环体至少执行一次](#)
- 13、C 语言中打开一个数据文件的系统函数为 ()。>[D. fopen \(\)。](#)
- 14、C 语言中的空白符是一些字符的统称，这些字符是>[A.空格、制表符、回车](#)
- 15、C 语言中的跳转类语句有四条，它们是 goto,return 和>[C.break,continue](#)
- 16、C 语言中的系统函数 fopen () 是 () 一个数据文件的函数。>[D. 打开](#)
- 17、C 语言中的选择类语句有两条，它们是 ()。>[A.if 和 switch](#)
- 18、C 语言中的循环类语句有三条，它们是 () >[B.for,while,do](#)
- 19、C 语言中函数返回值的类型是由 () 决定的。>[A. 函数定义时指定的类型](#)
- 20、C 语言中普通整型变量 int 在内存中占 () 字节。>[B. 2](#)
- 21、C 语言中使用的字符常量，其起止标记符是 ()。>[A.单引号](#)

- 22、C 语言中使用的字符串常量，其起止标记符是>[B.双引号](#)
- 23、C 语言中用于结构化程序设计的 3 种基本结构是 () >[A. 顺序结构、选择结构、循环结构](#)
- 24、C 语言属于 () 语言。>[C. 高级语言](#)
- 25、float 类型的长度为 ()。>[C.4](#)
- 26、for 循环语句“for (i=0;i<n;i+=2) S;”中循环体 S 语句被执行的次数为 ()。>[D.\(n+1\)/2](#)
- 27、for 循环语句“for (i=0; i<n; i+=2) S;”中循环体 S 语句被执行的次数为 ()。>[A. \(n+1\) /2](#)
- 28、for 循环语句能够被改写为 ()。>[D.while 语句](#)
- 29、for 循环语句能够被改写为 () 语句。>[D.while](#)
- 30、int 类型的长度为 ()。>[C.4](#)
- 31、NULL 是一个符号常量，通常作为空指针值，它的取值为 ()。>[A.0](#)
- 32、while 循环执行的次数是 () >[A.无限次](#)
- 33、x<0&&y<=5 的相反表达式为 ()。>[A.x<=0 || y>5](#)
- 34、x<=10 的相反表达式为 ()。>[B.x>10](#)
- 35、x>0&&x<=10 的相反表达式为 ()。>[A. x=0x10](#)
- 36、x>0&&y<=5 的相反表达式为>[A.x=0y5](#)
- 37、x>5 的相反表达式为 ()。>[D.x=5](#)
- 38、x<=10 的相反表达式为 ()。>[B.x>10](#)
- 39、保留字 case 和 default 所使用的语句是>[C.switch](#)
- 40、被调函数调用结束后，返回到： ()。>[D. 主调函数中该被调函数调用语句的下一语句](#)
- 41、标准输出设备显示器的文件流标识符是 ()。>[D.stdout](#)
- 42、标准输入设备的文件流标识符是 ()。>[B.stdin](#)
- 43、标准输入设备键盘的文件流标识符是 ()。>[D.stdin](#)
- 44、标准整数类型的长度为 () >[C.4](#)
- 45、表达式 floor (2.3457*100+0.5) /100 的值为>[A.2. 35](#)
- 46、表达式 floor (2.3457*100+0.5) /100 的值为 ()。>[A.2. 35](#)
- 47、表达式 floor (2.3457*100) /100 的值为>[C.2. 34](#)
- 48、表示文件结束符的符号常量 EOF 的值为 ()。>[C.-1](#)
- 49、表示文件结束符的符号常量为 ()。>[C.EOF](#)
- 50、不符合 C 语言规定的复合语句是 ()。>[B.{y=10}](#)
- 51、常数 120 的数据类型为>[A.整型](#)
- 52、常数 120 的数据类型为 ()。>[D.整型](#)
- 53、程序的运行结果是： ()。
#include<stdio.h>
inttest (intb)
{b*=10;
printf (“b=%d”,b) ;
returnb/2;
voidmain ()
{intb=60;B=test (b) ;
printf (“b=%d”,b) ;
}
- 程序的运行结果是： ([D.b=600b=300](#))。
- 54、程序运行中需要从键盘上输入多于一个数据时，各数据之间默认使用 () 符号作为分隔符。>[D.空格或回车](#)

- 55、程序运行中需要从键盘上输入多于一个数据时，若不特别规定分隔符，则输入的各数据之间所使用的分隔符为 ()。>[D. 空格或回车](#)
- 56、程序运行中需要从键盘上输入多于一个数据时，各数据之间默认使用 () 符号作为分隔符。>[D. 空格或回车](#)
- 57、程序运行中需要从键盘上输入多于一个数据时，各数据之间所使用的分隔符为 ()。>[D. 空格或回车](#)
- 58、程序运行中需要从键盘上输入多于一个数据时，若不特别规定分隔符，则输入的各数据之间所使用的分隔符为 ()。>[C.空格或回车](#)
- 59、从一个二进制文件中读取数据的系统函数为>[A.fread \(\)。](#)
- 60、从一个二进制文件中读取数据的系统函数为 ()。>[D.fread\(\)](#)
- 61、从一个数据文件中读人以换行符结束的一行字符串的函数为 ()。>[B.fgets \(\)。](#)
- 62、从一个文本文件中读取一个字符的系统函数为 ()。>[C.fgetc\(\)](#)
- 63、从一个文本文件中读取以换行符结束的一个字符串的系统函数为 ()。>[B.fgets\(\)](#)
- 64、带有随机函数的表达式 rand () %20 的取值范围是 ()。>[C.0~19](#)
- 65、带有随机函数的表达式 rand () %21 的取值范围是 ()。>[D.0~20](#)
- 66、带有随机函数调用的表达式 rand () %20 的值在 () 区间内。>[C. 0~19](#)
- 67、带有随机函数调用的表达式 rand () %50 的值在 () 区间内。>[D.1~49](#)
- 68、当不需要函数返回任何值时，则所使用的返回类型为 ()。>[C.void](#)
- 69、当处理特定问题时的循环次数已知时，通常采用 () 循环来解决。>[A. for](#)
- 70、当处理特定问题时的循环次数已知时，通常采用的语句是 ()。>[A. for](#)
- 71、当处理问题时的循环次数是确定的，则采用的最合适的循环方式为>[A.for](#)
- 72、当处理一个问题时的循环次数已知时，最好采用 () 循环来解决。>[A.for](#)
- 73、动态存储分配是在 () 对程序和数据进行存储分配。>[B.计算机运行过程中](#)
- 74、短整数类型的长度为 ()。>[B.2](#)
- 75、对二维数组的正确定义是： ()。>[C. inta\[\]\[3\]={1,2,3,4,5,6};](#)
- 76、对以下哪种 printf 的用法正确 ()。>[D.printf \(“%f”,4\);](#)
- 77、对于每个进行动态存储分配的函数，其返回值类型为>[D.void*](#)
- 78、对于无返回值的函数，进行函数定义时使用的函数类型关键字为>[B.void](#)
- 79、对于一个长度为 n 的字符串，保存它至少需要占用的存储字节数为 ()。>[C.n+1](#)
- 80、对于一个长度为 n 的字符串，保存它至少需要占用的存储字节数为 ()。>[C.n+1](#)
- 81、对于一个二维字符数组 a[M][N]，存储每个字符串的长度至多为 ()。>[C.N-1](#)
- 82、对于一维字符数组 a[N]，用来存储一个字符串的长度至多为>[D.N-1](#)
- 83、关于 C 语言文件叙述错误的是 () >[A.C 语言文件由记录组成](#)

84、关于递归定义的函数，下列说法正确的是（）。>**C.有些递归定义的函数可以“迭代计算”，有些递归定义的函数则必须“递归计算”**；

85、关于二进制文件和文本文件描述正确的为（）。>**D.一般中间结果数据需要暂时保存在外存上，以后又需要输入内存的，常用文本文件保存；**

86、函数 pow(4,3) 的值为（）。>**C.64**

87、函数 sqrt(9) 的值为>**B.3. 0**

88、函数调用 func(2+x,4*y,5) 中所含实参的个数为>**B.3**

89、函数定义参数表中的每个形参变量，其作用域范围为所在的整个（）。>**D.函数**

90、假定 a 是一个数组名，则数组元素 a[i] 的指针访问方式为>**D.*(a+i)**。

91、假定 a 为一个数组名，在下面的表达式中，存在语法错误的是（）。>**B.*a++**

92、假定 a 为一个数组名，则下面表达式中错误的是（）。B.*a++

93、假定 a 为一个数组名，则下面表达式中错误的是（）。>**B.*a++**

94、假定 a 为一个数组名，则下面存在错误的表达式为（）。>**B.*a++**

95、假定 a 为一个数组名，则下面存在错误的表达式为的（）。>**B.*a++**

96、假定 a 为一个整数类型的数组名，整数类型的长度为 4，则元素 a[4] 的地址比 a 数组的首地址大（）个字节。>**C.16**

97、假定 a 为一个字符数组名，则元素 a[i] 的指针访问方式为（）。>**B.*(a+i)**。

98、假定 i 的初值为 0，则在循环语句“while(i<n)(s+=i*i; i++;)”中，其循环体被执行的次数为（）。>**B.n**

99、假定 k 是一个 double 类型的变量，则定义变量 p 的正确语句为>**D.double*p=&k**

100、假定 k 是一个 double 类型的变量，则定义变量 p 的正确语句为（）。>**D.char*p="Thankyou!";**

101、假定 n 的值为 5，则表达式 n++ 的值为（）。>**B.5**

102、假定 n 的值为 6，则表达式 ++n 的值为（）。>**D.7**

103、假定 n 的值为 6 则表达式 n++ 的值为（）。>**A.6**

104、假定 p 是一个指向 double 类型的数据指针，则 p+1 所指的数据的地址比 p 所指的数据的地址大（）个字节。>**C.4**

105、假定 p 是一个指向 float 类型的数据指针，则 p+1 所指的数据的地址比 p 所指的数据的地址大（）个字节。>**B.2**

106、假定 p 是一个指向 float 型数据的指针，则 p+1 所指数据的地址比 p 所指数据的地址大>**C.4 个字节**

107、假定 p 是一个指向 float 型数据的指针，则 p+1 所指数据的地址比 p 所指数据的地址大（）个字节。>**C. 4**

108、假定 p 是一个指向 float 型数据的指针，则 p+1 所指数据的地址比 p 所指数据的地址增加的字节数为（）。>**C. 4**

109、假定 p 是一个指向数据对象的指针，则通过 p 表示该数据对象为（）。>**D.*p**

110、假定 p 是一个指针变量，则该变量的地址表示为（）。>**D.&p**

111、假定 p 所指对象的值为 25，p+1 所指对象的值为 42，则表达式 *p++ 的值为（）。>**C.25**

112、假定 p 所指对象的值为 25，p+1 所指对象的值为 42，则执行 *p++ 运算后，p 所指对象的值为>**B.42**

113、假定 p 所指对象的值为 25，p+1 所指对象的值为 42，则表达式 *p++ 的值为（）。>**A.25**

114、假定 p 为指向二维数组 intd[4][6] 的指针，则 p 的类型为>**A.int (*) [6]**

115、假定 s 被定义为指针类型 char* 的变量，初始指向的字符串为“Helloworld!”，若要使变量 P 指向 s 所指向的字符串，则 P 应定义为（）。>**A.cha*P=S;**

116、假定 x=4. 6，则函数 ceil(x) 的值为（）。>**D.5**

117、假定 x=4. 6，则函数 floor(x) 的值为>**A.4. 0**

118、假定 x 的值为 4，y 的值为 6，则表达式 x++*++y 的值为（）。>**B.28**

119、假定 x 的值为 5,y 的值为 6，则表达式 x++*--y 的值为（）。>**A.25**

120、假定 x 是一个逻辑量，则 (x&&!x) 和 (x||!x) 的值分别为（）。>**C.0 和 1**

121、假定变量 m 的定义为“intm=7;”，则下面语句格式正确的是（）。B.*a++

122、假定变量 m 定义为“intm=7;”，则下面定义语句格式正确的是（）。B.int*p=&m;

123、假定的总次数为（）的初值为（），则在循环语句“while(i<n)(s+=i*i; i++;)”中循环体被执行>**B.n**

124、假定对二维数组 a[3][4] 进行初始化的数据为{{3, 5, 6}, {2, 8}, {7}}, 则 a[1][1] 被初始化为（）。>**D.8**

125、假定对二维数组 a[3][4] 进行初始化的数据为，则 a[1][1] 被初始化为>**D.8**

126、假定二维数组的定义语句为“chara[M][N];”，则该数组所含元素的个数为>**C.M*N**

127、假定二维数组的定义语句为“int*a[3][5];”，则该数组所占用的存储空间的字节数为（）。>**D.60**

128、假定二维数组的定义语句为“inta[3][5];”，则该数组所含元素的个数为（）。>**B.15**

129、假定二维数组的定义语句为“int*a[3J][5];”，则该数组所占用的存储空间的字节数为（）。D. 60

130、假定是一个指向 float 型数据的指针，则 p+1 所指数据的地址比所指数据的地址大（）。C. 个字节

131、假定要访问一个结构指针变量 x 所指对象中的数据成员 a，则表示方法为（）B.x->a

132、假定要访问一个结构 x 中的由 a 指针成员所指的对象，则表示方法为（）。>**B.x.a**

133、假定要访问一个结构变量 x 中的数据成员 a，则表示方法为（）。>**A.x. a**

134、假定要访问一个结构变量 x 中的由指针成员 a 所指的对象，则表示方法为（）。>**D.*(X. a)**。

135、假定要访问一个结构指针 p 所指对象中的 b 指针成员所指的对象，则表示方法为>**C.*(p-b)**。

136、假定要访问一个结构指针变量 x 中的数据成员 a，则表示方法为（）B.x->a

137、假定一堆数组的定义为“char.a[8];”，则数组所占用的存储空间的字节数为（）。D. 32

138、假定一个磁盘数据文件占用 n 个字节的存储空间，则按字节进行编址的范围是（）。>**B.0~(n-1)**

139、假定一个二维数组 a 的定义为“inta[3][4]={{3,4},{2,8,6}};”，则元素 a[2][0] 的值为（）。>**A.0**

140、假定一个二维数组的定义为“inta[3][4]=《3,4》,(2,8,6)”;”，则元素 a[1][0] 的值为（）。>**B.2**

141、假定一个二维数组的定义语句为“inta[3][4]={{3,4},{2,8,6}};”，则元素 a[1][2] 的值为（）。>**D.6**

142、假定一个二维数组的定义语句为“inta[3][4]={{3,4},{2,8,6}};”，则元素 a[1][1] 的值为（）。>**D. 8**

143、假定一个二维数组的定义语句为“inta[3][4]={{3,4},{2,8,6}};”，则元素 a[1][2] 的值为（）。>**C. 6**

144、假定一个二维数组的定义语句为“inta[3][4]={{3,4},{2,8,6}};”，则元素 a[2][1] 的值为（）。>**A. 0**

145、假定一个函数的参数说明为 constinta，则在函数体中不能进行的操作是>**D.a=10**

146、假定一个函数的二维数组参数说明为 charw[][N]，与之等价的指针参数说明为（）。>**A.char>(*w)[N]**

147、假定一个函数的数组参数说明为 chara[]，与之等价的指针参数说明为（）。>**B.char*a**

148、假定一个函数的原型为 intff(intx)，一个整型变量为 a，则下面函数调用表达式不正确的是（）。>**C.ff(*a)**

149、假定一个函数的原型语句为“intff(int.x)”; “，一个整型数组为 a[10]，则下面函数调用表达式不正确的是（）。B.ff(a[0])

150、假定一个函数定义为“externchar*f1(char*x){returnx;}”，表示该函数作用域的关键字为（）。>**D.extern**

151、假定一个函数定义为“externchar*f1(char*x){returnx;}”，表示该函数作用域的关键字为>**A.extern**

152、假定一个函数定义为“externchar*f1(char*x){returnx;}”，该函数返回值类型为>**B.char***

153、假定一个函数定义为“staticintf1(intx,inty){returnx+y;}”，该函数名称为（）。>**C. f1**

154、假定一个函数原型为“char*func(intn)”，则该函数的返回值类型为（）。>**D.char**

155、假定一个函数原型为“char*func”，则该函数的返回类型为（）。B.int*>**D.char***

156、假定一个函数原型为“voidff(inta[],intn)”，则对应的函数指针类型为（）。>**C.void(*ff)(int*a,int)**

157、假定一个函数原型为 intf1(inta[][N],intn)，与数组参数等价的表示为（）。>**D.int(*a)[N]**

158、假定一个函数原型语句为“intf1(inta[],intn);”，与数组参数等价的表示为（）。>**D.int*a**

159、假定一个函数原型语句为“intf1(inta[], intn);”，与数组参数等价的表示为（）D.int*a

160、假定一个结构类型的定义为“structA{inta,b;doublec;};”，则该类型的长度为（）。>**A.16**

161、假定一个结构类型的定义为“structA{inta,b; doublec;};”，则该类型的长度为（）。>**D. 16**

162、假定一个结构类型的定义为“structB{inta[5];char*b;};”，则该类型的长度为（）。>**D.24**

163、假定一个结构类型的定义为“structD{inta; D*next;};”，则该类型的长度为（）。>**B. 8**

164、假定一个联合类型的定义为“unionD{inta;D*next;}”, 则该类型的长度为()。>[D.4](#)

165、假定一个链表的表头指针为f, 结点中包含有data和next域, 则向该链表的表头插入一个地址为p的结点时, 应执行的操作为p->next=f和>[C.f=p](#)

166、假定一个链表的表头指针为f, 结点中包含有data和next域, 则向该链表的表头插入一个地址为p的结点时, 应执行的操作为()。>[B. p-next=f和f=p](#)

167、假定一个链表中结点的结构类型为“structAA{intdata,structAA*next;}”, 则next数据成员的类型为()。>[D.structAA*](#)

168、假定一个链表中结点的结构类型为“structAA{intdata,structAA*next;}”, 并且有定义语句为“structAAa,b={3,&a}”, 则下面说法正确的是>[D.a是b的后继结点](#)

169、假定一个链表中结点的结构类型为“structAA{intdata,structAA*next;}”, 则next数据成员的类型为>[B.structAA*](#)

170、假定一个枚举类型的定义为“enumRA{ab,ac,ad,ae}”, 则ac值为>[C.1](#)

171、假定一个枚举类型的定义为“enumRB{ab,ac=3,ad,ae}”, 则ad的值为>[B.4](#)

172、假定一个数据对象为int*类型, 则指向该对象的指针类型为>[C.int**](#)

173、假定一维数组的定义为“char*a[8]”, 则该数组所占用的存储空间的字节数为()。>[D.32](#)

174、假定有“structBOOK{chartitle[40];floatprice;}book;”, 则正确的语句为()。>[B. structBOOK*x=&book;](#)

175、假定有“structBOOK{chartitle[40];floatprice;}structBOOKbook;”, 则不正确的语句为()。>[A. structBOOK*x=malloc\(book\)。](#)

176、假定有定义“intb[10],*pb;”, 则不正确的赋值为()。>[B.pb=b\[5\]](#)

177、假定有定义“inta[10],x,*pa=a;”, 若要把数组a中下标为3的元素值赋给x, 则不正确的赋值为()。>[D.x=*pa+3](#)

178、假定有定义“inta[4][5]={1,3,5},{2,8}”, 则a[1][1]的值为()。>[D.8](#)

179、假定有定义“intm=7,*p;”, 则给p赋值的正确表达式为()。>[B. p=&m](#)

180、假定有结构体定义“structBook{chartitle[20];doubleprice;}”, 则不正确的语句定义为()。>[B.struct Book b=\(“C++ Programming”,27, 0\)%struct Book a\[5\]%struct Book d\[2\]\[3\]](#)

181、假定有一个int型一维数组a[10], 则元素a[5]的字节地址为>[D.\(char*\)a+20](#)

182、假定有一个定义语句为“inta[10]={3,4,2,8,6}”, 则元素a[3]的值为()。>[D.8](#)

183、假定有一个定义语句为“inta[10]={3,4,2,8,6}”, 则元素a[5]的值为()。>[A.0](#)

184、假定有一个定义语句为“inta[10]={3,4,2,8,6}”, 则元素a[3]的值为>[C.8](#)

185、假定有一个定义语句为“inta[10]={3,4,2,8,6}”, 则元素a[5]的值为>[A.0](#)

186、假定有一个定义语句为“inta[10]=(3,4,2,8,6);”, 则元素a[3]的值为()。>[C.8](#)

187、假定有一个定义语句为“inta[3][4]={3,4},{2,8,6}”, 则元素a[1][2]的值为()。>[A.6](#)

188、假定有一个定义语句为“inta[3][4]=;”, 则元素a[1][2]的值为>[C.6](#)

189、假定有一个定义语句为“inta[3][4]=;”, 则元素a[2][1]的值为>[A.0](#)

190、假定有一个定义语句为“inta[10]={3,4,2,8,6};”, 则元素a[1]的值为() [A.4](#)

191、假定有一个定义语句为“inta[10]=(3,4,2,8,6);”, 则元素a[5]的值为()。>[A.0](#)

192、假定有一个定义语句为“inta[10]=(3,4,2,8,6);”, 则元素a[3]的值为() [C.8](#)

193、假定有一个元素类型为int的数组a[10], 它所占用的存储空间的字节数为()。>[D.40](#)

194、假定有一条定义语句为“inta[10]={3,4,2,8,6};”, 则元素a[5]的值为()。>[A.0](#)

195、假定有一条定义语句为“intm=7,*p;”, 则给p赋值的正确表达式为()。>[B.p=&m](#)

196、假定有语句“intb[10], *pb;”, 则不正确的赋值为()。>[B.pb=b\[5\]](#)

197、假定有语句为

“int *p=calloc(10+20,sizeof(int));”, 则P

所指向的动态数组中所包含的元素个数为()。

答: [C.30](#)

209、利用一维字符数组存储一个字符串时, 在其末尾自动存储的一个字符是()。>[B.'0'](#)

210、流程控制类语句包括的3类语句为()。>[C.选择、循环、跳转](#)

211、逻辑表达式(x>0&&x<=10)的相反表达式为()。>[C.“x<=0 && x>10”](#)

212、逻辑表达式(x>0||y==5)的相反表达式为()。>[B.“x<=0 && y!=5”](#)

213、逻辑表达式(x>0&&x<=10)的相反表达式为()。
[A. x<=0 || x>10](#)

214、枚举类型中的每个枚举常量的值都是一个()。>[A.整数](#)

215、每次先进行循环条件的判断, 然后再执行循环体的语句为()。>[A.for和while](#)

216、每个C程序文件在编译时可能出现有警告性错误, 其对应的标识符为()。>[A.warning](#)

217、每个C程序文件在编译时可能出现有严重性错误, 其对应的标识符为()。>[B.error](#)

218、每个C语言程序文件的编译错误分为()类。>[B.2](#)

219、能把函数处理结果的两个数据返回给主调函数, 在下面的方法中不正确的是()。>[A. return这两个数](#)

220、判定逻辑值为“真”的最准确叙述是()。>[C.非0的整数](#)

221、如果inta=3,b=4; 则条件表达式“a<b? a: b”的值是()。>[A.3](#)

222、如果一个函数定义仅允许为所在的程序文件调用, 则应在该函数定义的最前面添加的关键词为()。>[B.static](#)

223、如果一个函数作为表达式被调用, 则该函数必须是()。>[A.有返回值的函数](#)

224、若“intn; floatf=13.8;”, 则执行“n=(int)f%3”后, n的值是()。>[A.1](#)

225、若i的值为0, 则执行“do{++i; while(i<10);”语句的过程中, 循环体被执行的次数为>[A.4](#)

226、若n的值为10, 则执行“for(i=0; i<n/2) break;”语句后, i的值为()。>[6](#)

227、若n的值为10, 则执行“for(i=0; i<D.n(n+1)/2”语句后, i的值为>[B.6](#)

228、若n的值为10, 则执行“for(i=0; i<n/2) break;”语句后, i的值为>[C.4](#)

229、若n的值为10, 则执行“for(i=0; ; i++) if(i>n/3) break;”语句后, i的值为>[C.4](#)

230、若n的值为10, 则执行“for(i=1; i<=n; i++) for(j=1; j<=n; j++) S;”语句后, 内层循环体S语句的执行总次数为>[A.n*n](#)

231、若n的值为10, 则执行“for(i=1; i<n; i++) S;”语句的过程中, 表达式i=1共被执行的次数为()。>[A.1](#)

232、若n的值为10, 则执行“while(n>0)s+=n;”语句的过程中, 循环体被执行的次数为>[D.10](#)

233、若n的值为5, s的值为0, 则执行“dos+=-n; while(n>0);”语句后, s的值为>[D.10](#)

234、若n的值为5, s的值为0, 则执行“dos+=n--; while(n>0);”语句后, s的值为>[A.15](#)

235、若n的值为5, s的值为0, 则执行“while(n>0)s+=n--;”语句后, s的值为()。>[A.15](#)

236、若 n 的值为 5, s 的值为 0, 则执行“while (n>0) s+=n;”语句后, s 的值为>D.10

237、若 P 指向 x, 则与的等价表示为 ()。B. 个字节

238、若 p 指向 x, 则与 x 的等价表示为>B.*p

239、若 x=5, y=10, 则计算 y*=++x 表达式后, y 的值为 ()。>B.60

240、若 x 的值为-10, 则执行“if (!x) x++; else--;”语句后, x 的值为>D.-11

241、若 x 的值为-10, 则执行“if (x) x++;else--;”语句后, x 的值为 ()。>C.“-9”

242、若 x 的值为-10, 则执行“if (x) x++; else--;”语句后, x 的值为>B.-9

243、若 x 的值为 10, 则执行“switch (x%3) {case0: x++; break; case1: x++; default: x++;}”语句后, x 的值为>D.12

244、若 x 的值为 10, 则执行“switch (x%3) {case0: x++; case1: x++; case2: x++;}”语句后, x 的值为>C.12

245、若 x 的值为 15, 则执行“if (x>10) x++; else--;”语句后, x 的值为>A.16

246、若 x 的值为 5, 则执行“if (x>10) x++; else--;”语句后, x 的值为>C.4

247、若 x 的值为 9, 则执行“switch (x%3) {case0: x++; break; case1: x++; default: x++;}”语句后, x 的值为>B.10

248、若需要把一个字符串赋给字符数组, 则选用的函数是 ()。>A.strcpy

249、若需要比较两个字符串的大小, 则选用的函数是 ()。>C.strcmp

250、若需要从一个字符串中查找另一个子串, 则选用的函数是>B.strstr

251、若需要从一个字符串中查找一个字符, 则选用的函数是>C strchr

252、若需要定义一个符号常量, 并且使 C 语言能够进行类型检查, 则应在定义语句的开始使用保留字 ()。>A.const

253、若要求出一个字符串的长度, 则选用的函数是>A.strlen

254、若要把一个整型指针 p 转换为字符指针, 则采用的强制转换表达式为 ()。>A.(char*)p

255、若要定义整型指针 p 并初始指向 x, 则所使用的定义语句为 ()。>C.int *p=&x;

256、若要结束所在函数的执行过程, 返回到调用该函数所在位置, 则使用的语句为 ()。>C.return

257、若要使 p 指向二维整型数组 a[10][20], 则 p 的类型为 ()。>D.int (*) [20]

258、若要使 p 指向二维整型数组 a[10][20], 则 p 的类型为 ()。>A.int(*)[20]

259、若要读和写两种操作方式打开一个二进制文件, 当文件不存在时返回打开失败信息, 则选用的打开方式字符串为 ()。>C.“rb+”

260、若一个函数声明为“double*function () ;”, 则该函数的返回值类型为>B.实数指针型

261、若以下选项中的变量已正确定义, 则正确的赋值语句是 ()。>C. x3=0*12;

262、若用数组名作为函数调用的实参, 传递给形参的是 ()。>A.数组的首地址

263、若有定义: int x=0, *p=&x; , 则语句 printf (“%d/n“, *p); 的输出结果是 ()。>B.0

264、若有说明: int a[[4]]=[0, 0]; 则下面不正确的叙述是 ()。>D.只有元素 a[0]和 a[0][1]可得到初值 0, 其余元素均得不到初值 0

265、若有说明语句“inta[5], *p=a;”, 则对数组元素的正确引用是: ()。>C. *(p+2)

266、若有一个函数原型为“double*function ()”, 则其返回值类型为 ()。>B.实数指针型

267、若有一个函数原型语句为“dCble-funCion ()”; “, 则此函数的返回值类型为 ()。>C.函数指针型

268、若有以下数组说明, 则 i=10; a[a[i]]元素数值是 ()。>C. 6

269、若有语句为“inta[10], x, *pa=a;”, 要把数组 a 中下标为 3 的元素值赋给 x, 则不正确的语句为 ()。>D. x=*pa+3;

270、设 x 和 y 均为逻辑值, 则 x&& y 为真的条件是 ()。>A.它们均为真

271、设 x 和 y 均为逻辑值, 则 x&& y 为真的条件是 x 和 y ()。>B.均为真

272、设 x 和 y 均为逻辑值, 则 x||y 为假的条件是 ()。>C.它们均为假

273、设有变量说明 int (*p) [m], 其中的标识符 p 是 ()。>C.一个指向具有 m 个整型元素的一维数组的指针

274、设有两条语句为“ina=3; a+=a*a;”, 则执行结束后, a 的值为 ()。>A.12

275、设有两条语句为“inta=12; a+=a*a;”, 则执行结束后, a 的值为 ()。>C. 156

276、设有语句序列为“inta=12; a+=a*a;”, 则执行结束后, a 的值为 ()。>C.156

277、十进制数 25 表示成符合 C 语言规则的八进制数为>A.031

278、十进制数 25 表示成符合 C 语言规则的八进制数为 ()。>31

279、十进制数 25 表示成符合 C 语言规则的十六进制数为>B.0x19

280、十进制数 50 表示成符合 C 语言规定的八进制数为 ()。>D.062

281、十进制数 65 表示成符合 C 语言规定的八进制数为 ()。>D.101

282、实数 340. 25 对应的规格化浮点数为>C.3. 4025e2

283、使用“typedef int ABC[10];”语句把标识符 ABC 定义为 ()。>整型数组类型

284、使用“typedef int Integer;”语句把标识符 Integer 定义为一个 ()。>整数类型

285、数值常量 0x62 的表示方式为>C.十六进制

286、算数运算符, 赋值运算符, 关系运算符的优先级按从高到低依次为 ()。>B.算数运算符, 关系运算符, 赋值运算符

287、为了避免嵌套的 if-else 语句的二义性, C 语言规定 else 总是与 () 组成配对关系。>C. 在其之前最近的未配对的 if

288、无论在何种操作系统及 C 程序开发环境下, 运行 C 程序都必须按顺序经过 () 三个步骤之后程序才能运行。>B. 编辑编译链接

289、下列表达式中, 与下标引用 a[k]等效的是: ()。>A. *(a+k)

290、下列表达式中的最大值是>B.sizeof (double)。

291、下列不属于 C 语言中使用的常量的是 ()。>D.标识符常量

292、下列各表达式中, 结果为字符型的表达式为 ()。>(char)36

293、下列各语句定义了数组, 其中哪一个是不正确的 ()。>C. floatx[2][i]={1,2,4,6,8,10};

294、下列关于 break 语句的描述中, 不正确的是: ()。>C. eak 语句可用于 if 体内, 它将退出 if 语句

295、下列关于 C 语言的说法错误的是: ()。>B. C 语言不区分大小写。

296、下列关于变量作用域的描述中, 不正确的是 ()。>C.类变量能在类的方法中声明

297、下列关于函数定义的说法, 正确的是 ()。>B.函数定义不可以嵌套, 但可以嵌套调用

298、下列关于结构体的说法, 哪一项是错误的? ()。>D.结构体变量不能直接相互赋值

299、下列关于转义字符说法正确的是 ()。>D.转义字符 '\n' 功能是换行

300、下列结论中, 只有 () 是不正确的。>D. C 语言所有函数都可以相互调用, 包括调用 main。

301、下列数值中最大值是>C.28%8

302、下列数值中最大值是 ()。>C.28%8

303、下列四组选项中, 正确的 C 语言标识符是 ()。>C. a123

304、下列一组初始化语句中, 正确的是 ()。>B. inta[8]={0,2,4,0,1}

305、下列运算符, 其优先级最高的是 ()。>A.[]

306、下列运算符, 其优先级最高的是 ()。>A.[]

307、下列指针的定义不正确的定义是 ()。>A. int*p=&i.i;

308、下面 () 表达式的值为 4。>D. (int) (11. 0/3+0. 5)

309、下面 C 语言语句定义了 ()。
struct student {intnum; Char name[20]; Char sex; }s1,s2;
答: D.结构类型 student 和结构变量 s1,s2"

310、下面的保留字不能作为函数返回类型的是>C.sizeof

311、下面的函数原型声明中存在语法错误的是 ()。
C. AA(int a; int b;)

312、下面的函数原型声明中存在语法错误的是 ()。>D.void AA(int a; int b);

313、下面的描述错误的是 ()。>C. 函数内可以嵌套定义函数

314、下面循环语句执行结束后输出的 i 值为 ()。
for(int i=0; i<n; i++) if(i>n/2) {cout<<i<<endl; break;}
答案: B.n/2+1

315、下面有关 typedef 的叙述中, 正确的是 ()。>C.typedef 用于给已定义类型取别名

316、下面运算符优先级最高的是 ()。>B.[]

317、下面正确的函数原型语句是>B.voidFunction (int)。

318、向一个二进制文件中写入数据的系统函数为>B.fwrite ()。

319、向一个二进制文件中写入信息的函数 fwrite () 带有的参数个数为 ()。>D. 4

320、向一个二进制文件中写入信息的函数为 ()。>D. fwrite ()

321、向一个文本文件中写入一个字符串的系统函数为>C.fputs ()。

322、向一个文本文件中写入一个字符的系统函数为 ()。>C.fputs ()

323、循环体至少被执行一次的循环语句为>C.do-while

324、循环体至少被执行一次的循环语句为 ()。>D.for

325、循环体至少被执行一次的循环语句为 ()。>do-while

326、循环语句“for (i=0; i<=n; i++) S;”中循环体 S 被执行的次数为 ()。>D.n+1

327、要定义 a 和 b 为两个整型变量,则使用的定义语句为>B.int a,b;

328、要结束任一种循环语句或 switch 语句的执行,则在其中使用的语句为>B.break

329、要结束所在循环语句中循环体的一次执行,则在其中使用的语句为 ()。>C.continue

330、要利用#define 命令定义 ABC 常量的值为 100,则对应的命令格式为>C.#define ABC100

331、要利用 const 保留字定义整型常量 ABC 的值为 10,则对应的定义语句为 ()。>D.const int ABC=10;

332、要使 p 指向二维整型数组 a[10][20],p 应定义为 ()。>int(*p)[20]=a;

333、要使以下程序段输出 10 个整数,请填写一个整数;for (i=0; i<= () ; printf (“%d\n”, i+2)) ; >B.18

334、要使字符串变量 str 具有初值“happy”,正确的初始化语句为: ()。>C.char str[]=”happy”;

335、一个 C 语言程序必须包含一个 () 函数。>B. main

336、一个 C 语言的源程序中 () >A. 必须有一个主函数

337、一个包含逗号的表达式为 (10,20,30),则该表达式的值为>C.30

338、一个程序文件开始使用的每条预处理命令,其首字符必须是>B.#

339、一个程序文件开始使用的每条预处理命令,其首字符必须是 ()。>C.“#”

340、一个程序中用户定义的普通函数的名称不能为 ()。>B. main

341、一个二维数组元素为 a[i][j],它的指针访问方式不正确的是 ()。>D.*a[i]+j

342、一个二维字符数组 a[M][N]能够存储的字符串个数至多为 ()。>A.M

343、一个函数定义所包含的两个部分是>A. 函数头和函数体

344、一个无符号字符类型的值域范围是 ()。>B.0 到 255

345、一个有符号字符类型的值域范围是 ()。>-128 到+127

346、一个指针类型的简单变量占用内存中的字节数为 ()。>C.4

347、一个指针指向一个数据对象,它保存着该数据对象的地址,若数据对象为 DataType 类型,则该指针的类型为 ()。>B.DataType*

348、已知“char*s=”Helloworld!”;”,要使变量 p 指向 s 所指向的同一个字符串,则 p 应定义为>A.char*p=s;

349、已知“char a[3][10]={”1234”,”3456”,”567890”};char (*p)[10]=a[1];”,则 p[0][1]的字符值为 ()。>4

350、已知“char a[3][5];”,则元素 a[2][3]的地址比该数组的首地址大 ()。>A.13 个字节

351、已知“char b[10]=”12345”,*p=b;”,则*(p+2)的值为>B.’3’

352、已知“int*p=calloc (24,sizeof (int));”,则 p 所指向的动态内存空间中,能够存储整型元素的个数为 ()。>A.24

353、已知“int*p=malloc (100);”,要释放 p 所指向的动态内存,正确的语句为>A.free (p)。

354、已知“inta[10],x,*pa=a;”,若要把数组 a[3]元素的值赋给 x,则不正确的语句为 ()。>C.x=*pa+3;

355、已知“inta[10],x,*pa=a;”,若要把数组 a[3]元素的值赋给 x,则不正确的语句为>D.x=*pa+3;

356、已知“inta[3][5]={ {2,3,4},{3,5,7},{4,6,9}};int (*p) [5]=a;”,则表达式 (*(p[1]+2)) 的值为 ()。>7

357、已知“inta[3][5]=; int (*p) [5]=a;”,则 p[1][1]的值为>B.5

358、已知“inta[3][5]=; int (*p) [5]=a;”,则表达式 (*(p[1]+2)) 的值为>C.7

359、已知“inta[3][5];”,则元素 a[2][3]的地址比该数组的首地址大 ()。>B.52 个字节

360、已知“intb[10];int*pb;”,则下面不正确的赋值语句为 ()。>pb=b[5];

361、已知“intb[10]={1,2,3,4,5,6},*p=b;”,则*(p+3)的值为>C.4

362、已知“ (hara[3][5]);”,则元素 a[2][3]的地址比该数组的首地址大 ()。>A.13 个字节

363、已知“A~Z”的 ASCII 码为 65~90,当执行“intx=C+5;”语句后 x 的值为 ()。>B.72

364、已知 inta[3][4];则对数组元素引用正确的是: ()。>C. a[2][0]

365、已知 x 的初值为 15,则下列各表达式中的最大值为 ()。>B.+x

366、已知 x 的初值为 15,则下列各表达式中的最大值为 ()。>x*=2

367、已知 x 的初值为 15,则下列各表达式中的最小值为 ()。>B.x/=2

368、已知 x 的初值为 5,则下列各表达式中的最大值为>D.x*=4

369、已知: inta[10];则对 a 数组元素的正确引用是 ()。>D. a[0]

370、已知“chara[3][5];”,则元素 a[2][3]的地址比该数组的首地址大 ()。>A.13 个字节

371、已知“chara[3][6];”,则元素 a[2][3]的地址比该数组的首地址大 () >A.15 个字节

372、已知“inta[10], x, *pa-a;”,若要把数组 a[5]元素的值赋给 x,则不正确的语句为 ()。>D.x=*pa+5;

373、已知“inta[10], x, *pa=a;”,若要把数组 a[3]元素的值赋给 x,则不正确的语句为 ()。>D.x=*pa+3;

374、已知“intb[10]; int*pb;”,则下面不正确的赋值语句为 () >D.pb=b[5]

375、已知“intp=malloc (100);”,要释放 p 所指向的动态内存,正确的语句为 ()。>A. free (p)。

376、已知“ (hara[3][5]);”,则元素 a[2][3]的地址比该数组的首地址大 () >A.13 个字节

377、已知数组 a 的定义为 chara[]=”abc\tden”;”,请写出 strlen (a) 的值为: ()。>A. 7

378、以下不能定义为用户标识符的是 () >D. scanf

379、以下对二维数组 a 的正确说明是 () >C.float a(3)(4);

380、以下对一维数组 a 的正确说明是: ()。>D.chara[3]={’a’,’b’,’c’};

381、以下关于循环的描述中,错误的是 ()。>D. do...while 语句与 while 语句的区别仅仅是关键字 while 的位置不同

382、以下关于循环体的描述中,错误的是: ()。>C. 循环体中不能出现 goto 语句

383、以下哪个不属于 C 语言的特点: ()。>D. 编写的程序不可移植

384、以下哪个是 C 语言的输出语句: ()。>C. printf

385、以下哪个是算法表示的方法: ()。>D. 以上均是

386、以下选项中,两个条件语句功能等价的是 ()。>C.if (a) printf(“%dn”,a); if (a==0) printf(“%dn”,a);

387、用 Calloc 函数创建具有 10 个整型元素的动态存储空间的正确调用是 ()

C. calloc(10,sizeof(int))

388、用 calloc 函数创建具有 10 个整型元素的动态存储空间的正确调用是 ()。>A.calloc(10,sizeof(int))

389、用 calloc 函数创建具有 10 个整型元素的一维数组的正确语句是 ()。>C. int*p=calloc (10,4)。

390、用 Callo 函数创建具有 10 个整型元素的动态存储空间的正确调用是 ()。>B.calloc (10)。

391、用于从键盘上为变量输入值的标准输入函数是 ()。>C.scanf()

392、用于输出表达式值的标准输出函数是 ()。>C.printf ()

393、由 C 语言目标文件连接而成的可执行文件的默认扩展名为 ()。>B.exe

394、由 C 语言目标文件连接而成的可执行文件的缺省扩展名为 ()。>B.exe

395、由 C 语言源程序文件编译而成的目标文件的缺省扩展名为: ()。>D. obj

396、有如下程序 inta[10]={1,2,3,4,5,6,7,8,9,10},*p=a;则数值为 9 的表达式是: (B.* (p+8))。

397、有以下程序段 intk=0;while (k=1) k++;while 循环执行的次数是 () >A.无限次

398、有语句: inta[10];则 () 是对指针变量 p 的正确定义和初始化。>B. int*p=a;

399、与结构成员访问表达式 p->name 等价的表达式为 ()。>A.(*p). name

400、与结构成员访问表达式 p->name 等价的表达式为 () >B.(*p).name

401、与结构成员访问表达式 x. name 等价的表达式为>C. (&x) -name

402、语句 int*p; 说明了 ()。>C. p 是指向 int 型数据的指针

403、运算符优先级最高的是>A. ()。

404、运算符优先级最高的是>B.||

405、运算符优先级最高的是 ()。>B.+

406、运算符优先级最高的是 ()。>D. !

407、运算符优先级最高的是 ()。>A.||

408、运算符优先级最高的是 ()。>B.++

409、运算符优先级最高的是 ()。>A. ()

410、在#include 预处理命令中,不可以包含 ()。>D.目标文件

411、在 C 语言程序中,表达式 5%2 的结果是: ()。>C. 1

412、在 C 语言程序中,单行或行尾注释语句使用的标记符为 ()。>C.//

413、在 C 语言程序中,多行注释语句使用的开始标记符为 ()。>B. /*

414、在 C 语言程序中,进行数据的输入和输出操作时,需要包含的系统头文件是>A.stdio. h

415、在 C 语言中,表示换行符的转义字符为>C.\n'

416、在 C 语言中,表示换行符的转义字符为 ()。>C.\n

417、在 C 语言中,打开一个数据文件的系统函数为 ()。>D.fopen()

418、在 C 语言中,每条复合语句的结束标记字符为 ()。>D. }

419、在 C 语言中,每条复合语句的开始标记字符为 ()。>C. ()。

420、在 C 语言中,每条复合语句的开始标记字符为 ()。>C. {

421、在 C 语言中，每条复合语句的结束标记字符为 ()。D. }
 422、在 C 语言中，所有预处理命令都是以 () 符号开头的。>B.#
 423、在 C 语言中，为读和写操作打开一个二进制文件，若文件不存在则自动建立空文件的打开方式为 ()。>A.wb+
 424、在 C 语言中，为只读操作打开一个文本文件的方式为>C.“r”
 425、在 C 语言中，为只写操作打开一个文本文件的方式为>D.“w”
 426、在 C 语言中，为追加写操作打开一个二进制文件的方式为 ()。>A.ab
 427、在 C 语言中，一条简单语句的结束符是>C.分号
 428、在 C 语言中，一条简单语句的结束符是 ()。>分号
 429、在 C 语言中，用户定义的标识符分为 4 种作用域，下列选项中不属于这 4 种作用域的是>B.语句
 430、在 C 语言中，用于比较不等于的关系运算符为>C.!=
 431、在 C 语言中，用于比较大小的关系运算符的总数为 ()。>D.6
 432、在 C 语言中，用于比较的关系运算符的总数为 ()。>6
 433、在 C 语言中，字符型数据在计算机内存中，以字符的 () 形式存储。>C. ASCII 码
 434、在 for 循环语句“for (i=0; i<n; i++) S;”中，循环体 S 被执行的次数为 ()。>A.n
 435、在 printf () 函数调用的格式字符串中，每个格式符的先导字符为>B.%
 436、在 printf () 函数调用的格式字符串中，每个格式符的先导字符为 ()。>A.“%”
 437、在 printf () 函数调用的格式字符串中，若使用格式符为“%5d”，则规定对应输出的数据占用的字符位置个数为 ()。>A. 5
 438、在 printf () 函数调用的格式字符串中，若使用格式符为“%6. 2f”，则规定对应输出的数据占用的字符位置个数为>B.6
 439、在 printf () 函数调用的格式字符串中，若使用格式符为“%6. 2f”，则规定对应输出的数据中，其整数部分占用的字符位置个数为 ()。>C.3
 440、在 printf () 函数调用的格式字符串中，若使用格式符为“%c”，则对应输出的数据类型为>A.char
 441、在 printf () 函数调用的格式字符串中，若使用格式符为“%d”，则对应输出的数据类型为 ()。>B.int
 442、在 printf () 函数调用的格式字符串中，若使用格式符为“%f”，则对应输出的数据类型为>C.float
 443、在 printf () 函数调用的格式字符串中，若使用格式符为“%Lf”，则对应输出的数据类型为>D.double
 444、在 printf () 函数调用的格式字符串中，若使用格式符为“%s”，则对应输出的数据为>C.字符串
 445、在 scanf () 函数调用的格式字符串中，每个格式符的先导字符为>B.%
 446、在 scanf () 函数调用的格式字符串中，每个格式符的先导字符为 ()。>B.@
 447、在 scanf () 函数调用的格式字符串中，每个格式符的先导字符为 ()。>B.%
 448、在 scanf () 函数调用的格式字符串中，若使用格式符为“%c”，则对应输入的数据类型为>A.char
 449、在 scanf () 函数调用的格式字符串中，若使用格式符为“%d”，则对应输入的数据类型为 ()。>int

450、在 scanf () 函数调用的格式字符串中，若使用格式符为“%f”，则对应输入的数据类型为>C.float
 451、在 scanf () 函数调用的格式字符串中，若使用格式符为“%Lf”，则对应输入的数据类型为 ()。>double
 452、在 scanf () 函数调用的格式字符串中，若使用格式字符串为“%d.%d”，则规定输入的两个数据之间的分隔符为>A.逗号
 453、在 scanf () 函数调用的格式字符串中，若使用格式字符串为“%d.%d”，则规定输入的两个数据之间的分隔符为 ()。>D.“@”
 454、在 switch 语句的每个 case 块中，假定都是以 break 语句结束的，则此 switch 语句容易被改写成的语句是 ()。>B. if
 455、在 switch 语句中，每个保留字 case 后面的表达式必须是>B.常量
 456、在 VC++ 环境下，为运行一个程序而建立的工作区文件的扩展名为 ()。>C.dsw
 457、在程序的一个文件中定义的函数，若要在另一个文件中调用，则必须在这另一个文件中给出该函数的 ()。>A.原型语句
 458、在定义指针变量的语句中，在指针变量名的前面标记的字符为>B.*
 459、在函数调用时，以下说法正确的是： ()。>B. 实际参数和形式参数可以同名
 460、在函数定义的参数表中，也可以使用函数参数，假定一个函数参数原型为 intff (int)，则等价的函数指针说明为 ()。>C.int (*ff)(int)
 461、在函数体中把出现调用自身函数的情况称为 ()。>B.直接递归调用
 462、在函数体中定义的变量具有>B.局部作用域
 463、在函数体中定义一个静态变量时，选用的存储属性关键字为>A.static
 464、在函数体中定义一个自动变量时，可以选用的存储属性关键字为>C.auto
 465、在函数外定义的变量，若不带有作用域关键字，则它具有 ()。>A.全局作用域
 466、在教材中所介绍的进行动态存储分配的函数有>C.3
 467、在结构类型的定义中，不同数据成员的定义项之间采用的分隔符错误的是 () >B.冒号
 468、在结构类型的定义中，不同数据成员的定义项之间采用的分隔符是 ()。>C.分号
 469、在结构类型的定义中，对于类型标识符后的定义体，使用的起止标记符是一对>B.花括号
 470、在结构类型的定义中，对于类型标识符后的定义体，使用的起止标记符是一对 ()。>A.圆括号
 471、在结构类型的定义中，使用的关键字是 ()。>B.struct
 472、在联合类型的定义中，使用的关键字是 ()。>C.union
 473、在每个 C 语言程序中都必须包含有这样一个函数，该函数的函数名为 ()。>A.main
 474、在下列的符号常量定义中，错误的定义语句格式为 ()。>C. constintM310;
 475、在下列的符号常量定义中，正确的定义格式为 ()。>C.“#define M3 10”
 476、在下列的语句中，必须以分号作为结束的语句是>C.do

477、在下列的字符数组定义中，存在语法错误的是 ()。>A.char a[10]=5;
 478、在下列选项中，不正确的函数原型格式为 ()。>C. intFunction (a);
 479、在下列语句中，正确的是>A.chars1[]="abcdef";
 480、在下面的 do-while 循环语句中，其循环体语句被执行的次数为 ()，inti=0; do1++; while (i<10)。>C.10
 481、在下面的 do-while 循环语句中，其循环体语句被执行的次数为 ()。>C. 10
 482、在下面的 do 循环语句中，其循环体被执行的次数为 ()。A.4
 483、在下面的 do 循环语句中，其循环体被执行的次数为 ()。inti=0; do1++; while (i*i<10); >4
 484、在下面的保留字中，不能作为函数的返回值类型的是 ()。>C. enum
 485、在下面的二维数组的定义语句中，语句格式正确的是>C.inta[[3]]=[1,3,5];
 486、在下面的二维数组定义中，语法正确的选项是 ()。>C. inta[[3]={{1,3,5},{2}}];
 487、在下面的函数声明语句中，存在语法错误的是>C.AA (inta; intb)。
 488、在下面的函数声明语句中，存在着语法错误的是 ()。>C. AA (inta; intb)。
 489、在下面的函数声明语句中存在着语法错误的是 ()。>A.AA(int a; int b)
 490、在下面的选项中，不能作为函数的返回值类型的是 ()。>B.node
 491、在下面的选项中，不能作为函数值返回类型的是 ()。>A.new
 492、在下面的一维数组定义中，错误的定义格式为 ()。>C. inta[1
 493、在下面的一维数组定义中，错误的定义为： ()。>C. chara[10]={0};
 494、在下面的一维数组定义中，有语法错误的选项是 ()。>C. inta[1];
 495、在下面循环语句中，内层循环体 S 语句的执行总次数为 ()。for(inti=0;i<n;i++) for(intj=i;j<n;j++)S;
 答案：D.n(n+1)/2
 496、在下面运算符中，优先级最高的是 ()。>A.“!”
 497、在下面字符数组的定义语句中，有语法错误的是 ()。>B.char a[10]=5;
 498、在循环语句“for (i-1; i<10; i+2) S;”中，循环体 S 被执行的次数为 ()。>D.5
 499、在循环语句“for (i=0;i ()。>n
 500、在循环语句“for (i=0, i<ni++) S, “中，假定 n 的整数值大于 0，则循环体 S 被执行的次数为 ()。>C.n
 501、在循环语句“for (i=1; i<10; i+=2) S;”中，循环体 S 被执行的次数为 ()。>D.5
 502、在循环语句“for (i=1; i<10; i+=3) S;”中，循环体 S 被执行的次数为 ()。>A.3
 503、在循环语句“for (i=n-1; i>=1; i--) S;”中，循环体 S 被执行的次数为>B.n-1

504、在循环语句“for (i=0; i<n; i++) S; “中，循环体 S 被执行的次数为 () C.n

505、在一个程序文件的开始，若需要使用一个“包含命令”包含一个头文件时，则该标识符为 >C.#include

506、在一个程序文件的开始，若需要使用一个包含命令包含一个头文件时，则命令标识符为 () 。>C.“#include”

507、在一个程序文件中，若要使用#include 命令包含一个系统头文件，则此头文件所使用的起止定界符为一对 () 。>A.尖括号

508、在一个程序文件中，若要使用#include 命令包含一个用户定义的头文件，则此头文件所使用的起止定界符为一对 () 。>B. 双引号

509、在一个函数定义中，函数头包括的 3 个部分依次为函数类型、函数名和 >D.参数表

510、在一个函数定义中，函数头包括的 3 个部分依次为函数类型、函数名和 () 。>C. 参数表

511、在一个链表中，每个结点必然存在着一个指向自身结点类型的指针域，用来保存下一个结点的 >A.地址

512、在一个数组中，数组元素 a[i] 前面的元素个数为 () 。>A.i

513、正弦函数 sin48 在 C 语言的表示为 () 。>D.sin(48*3.14159/180)

514、执行“int*p=malloc (10*4*sizeof (int)) ;”语句后，p 所指向的动态存储空间中包含有整型元素的个数为 >C.40

515、执行“int*p=malloc (sizeof (double) *2);”语句后得到的动态存储空间的字节数为 () 。>C.16

516、执行“printf (“%c”,B+2);”语句后得到的输出结果为 >D.D

517、转义字符“\”表示的字符是 () 。>C. 反斜线

518、字符串“a+b=12/n”的长度为 () 。>B. 7

519、字符串“\t/s/”中包含的字符个数为 >D.5

520、字符串“\t/s/40/n”的长度为 >A.5

521、字符串“y=%d/n”的长度为 () 。>A.5

522、字符串“y=%d/n”的长度为 () 。>A.5

523、字符串“y=%d/n”的长度为 () 。>A.5

524、字符串“y=%d/n”的长度为 () 。>A.5

525、字符串“y=%d/n”的长度为 () 。>A.5

526、字符串“y=%d/n”的长度为 () 。>A.5

527、字符串“y=%d/n”的长度为 () 。>A.5

528、字符串“y=%d/n”的长度为 () 。>A.5

529、字符串“y=%d/n”的长度为 () 。>A.5

530、字符串“y=%d/n”的长度为 () 。>A.5

9、对于 for (表达式 1;表达式 3) 下面理解错误的是 () 。>(A.for(表达式 1;0;表达式 3)C.for(表达式 1;表达式 1;表达式 3)D.for(表达式 1;表达式 3;表达式 3))

10、给出以下定义:charx[]={"abcdefg";chary[]={ 'a','b','c','d','e','f','g'}; 则不正确地叙述为 () A.数组 X 和数组 Y 等价 B.数组 x 和数组 Y 地长度相同 D.数组 X 地长度小于数组 Y 地长度

11、给定: inta,b;scanf (“%d%d",&a,&b); 则以下哪些输入方式是正确的? () 。>(A.1 2C.1(回车)2)

12、关于 C 语言文件操作的结论中，() 是错误的。>(A.对文件操作必须先关闭文件 C.对文件操作顺序无要求 D.对文件操作前必须先测试文件是否存在，然后再打开文件)

13、关于函数声明，以下正确的说法是 () 。>(A.如果函数定义出现在函数调用之前，可以不必加函数原型声明 B.如果在所有函数定义之前，在函数外部已做了声明，则各个主调函数不必再做函数原型声明 D.标准库不需要函数原型声明)

14、关于文件理解正确的为 () 。>(B.所谓文件一般指存储在外部介质上数据的集合; C.系统自动地在内存区为每一个正在使用的文件开辟一个缓冲区; D.每个打开文件都和文件结构体变量相关联，程序通过该变量访问该文件;)

15、假定要访问一个结构变量 x 中的由指针成员 a 所指向的对象，则表示方法不正确的是 () 。>(A.x-a B.X.a C.*(A.x))

16、假定有结构定义“structBook{char*title;doubleprice;};”，则语句定义错误的是 () >(A.struct Book *x=malloc (6,sizeofstruct (Book)) B.struct Book *x=malloc (6*sizeof (struct Book)) D.struct Book *x=malloc (6*sizeof (struct Book *))

17、若 a,b,c,d 都是 int 型变量且初值为 0,以下选项中正确的赋值语句是 () 。>(A.a=b=c=d=100;B.d++;D.d=(c=22)-(b++);)

18、若变量已正确定义并赋值，下面不符合 C 语言语法的表达式是 () 。>(A.a:=b+1C.int 18. 5%3D.a=a+7=c+b)

19、若定义: inta【2】【3】{0,2,4,6,8,10}; 以下描述正确的有 () 。>(A.* (a+1) 为元素 6 的地址 B.* (A【1】+1) 的值为 2C.** (a+1) +2 的值为 8D.a【0】与 a 的相同)

20、设 inta=3,b=4,c=5; 则下面的表达式中,值为 0 的表达式是 () 。>(A.a>b C.(a|b)>c D.a<=(b|c))

21、设 int (*p) [4]; 则 p 的含义是 () >(A.指向整型变量的指针数组 B.指向 4 个整型变量的函数指针 C.4 个指向整型变量的指针变量 D.指向具有 4 个整型元素的一维数组的指针)

22、下列 C 语言用户标识符中不合法的是 () 。>(A.3axC.caseD.0)

23、下列不正确的标识符是 () 。>(A.0B.afilD.int t)

24、下列关于 C 语言的说法正确的是 () 。>(A.C 程序的工作过程是编辑、编译、连接、运行 C.C 程序的三种基本结构是顺序、选择、循环 D.C 程序从 main 函数开始执行)

25、下列关于 C 语言地叙述正确地是 () 。>(B.不同类型地变量可以在一个表达式中 C.在赋值表达式中等号(=)左边地变量和右边地值可以是不同类型 D.同一个运算符在不同地场合可以有不同地含义)

26、下列关于变量作用域的叙述，正确的是 () 。>(A.变量只能在其作用域范围内访问 C.局部变量具有块作用域 D.全局变量具有文件作用域)

27、下列描述中正确地是 () 。>(A.字符型数组中可以存放字符串 B.可以对字符型数组进行整体输入、输出 D.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)

28、下列哪些属于数学函数 () 。>(A.double sin(double x);B.B.int rand(void);C.C.double floor(double x);D.E.double exp(double x);)

29、下列说法中不正确地是 () 。>(A.在程序中定义一个结构体类型，将为此类型分配存储空间 C.结构体类型必须有名称 D.结构体内地成员不可以是结构体变量;)

30、下列四组选项中，不是 C 语言标识符是 () 。>(A.%xB.a+bD.123)

31、下列叙述中，正确的是 () 。>(B.在其它函数中定义地变量在主函数中也不能使用 C.形式参数也是局部变量 D.复合语句中定义地变量只在该复合语句中有效)

32、下列选项中是 C 语言合法标志符的有 () 。>(A.good _morningB.main)

33、下列属于字面常量的是 () >(A.整型常量 B.字符常量 C.枚举常量 D.实型常量)

34、下面对 typedef 的叙述中正确的是 () 。>(A.用 typedef 可以定义各种类型名，但不能用来定义变量 C.用 typedef 只是将已存在的类型用一个新的标识符来代表 D.使用 typedef 有利于程序的通用移植)

35、下面关于 malloc 标准库函数的描述中，正确的是 () 。>(A.通过 malloc 函数可以在程序运行过程中分配存储 C.malloc 函数是返回空类型指针的函数 D.malloc 函数的参数是分配空间的字节数)

36、下面关于递归函数说法不正确的 () >(A.一般来说，递归函数的执行效率高于非递归函数 C.递归函数的嵌套调用次数没有限制 D.递归函数不可以改写为非递归函数)

37、下面关于结构体数据类型描述中，正确的是 () 。>(A.定义结构体类型时,系统会根据各个成员的大小为其分配内存空间 B.定义结构体类型时,系统会根据最大成员的大小为其分配内存空间 D.定义结构体类型时,系统会分配固定内存)

38、下面哪些是合法的用户定义的标识符 () 。>(A. 908D.wind)

39、下面哪些写法是对的,而且是常量? () 。>(B.2. 5E+21D.///'123)

40、下面说法中，正确地是 () 。>(A.全局变量一定是外部变量 B.局部变量一定是内部变量 C.全局变量作为函数间传递数据手段时，与文件地作用类似，都是通过共享某种资源来传递数据)

41、一个用户函数定义包含哪些部分 () >(B.函数名 C.函数体)

42、已知 a 为整型变量，那么与表达式 a!=0 真假值情况相同的表达式有哪些? () 。>(A.a>0|a<0B.a C.!a==0)

43、已知 inta;使用 scanf () 函数输入一个整数给变量 a,错误的的函数调用是 () 。>(A scanf(“%d”,a);B scanf(“%d",&a);D scanf(“%lf",&a);)

44、以下 4 个选项中,可以看作是一条语句的有 () 。>(A.{;} B.a=0,b=0,c=0;C.if(a>0;))

45、以下 C 语言程序说法中不正确的 () 。>(A.C 语言程序总是从第一个定义的函数开始执行 B.在 C 语言程序中，要调用的函数必须在 main() 函数中定义 D.C 语言程序中的 main() 函数必须放在程序的开始部分)

46、以下对 C 语言中的函数描述不正确的有 () 。>(A.可以嵌套定义,不可以嵌套调用 C.可以嵌套定义,也可以嵌套调用 D.嵌套定义和嵌套调用都不允许)

47、以下关于数组的描述中，正确的有（）>(B.构成字符数组的各个元素必须是字符类型的数据 C.若有定义 char a[]="hello";，则数组 a 的长度为 6)

48、以下合法的十六进制数是（）>(B.0Xabc.C.0x01)

49、以下哪些是有效的变量名？（）>(A.main D. _int)

50、以下说法中不正确的是（）>(A.C 语言的指针只能用来装变量的地址 B.C 语言的指针可以进行加、减、乘、除等算术运算 D.C 语言指针很难，一般人难以理解)

51、以下说法中不正确的是（）>(A.C 源程序可以直接运行产生结果 B.C 源程序经编译后才可直接运行产生结果 C.C 源程序经连接后才可直接运行产生结果)

52、以下叙述正确的是（）>(A.一个 C 源程序可由一个或多个函数构成 B.一个 C 源程序必须包含一个 main 函数 C.C 程序地基本组成单位是函数)

53、以下叙述中，不正确地（）>(A.调用 printf 函数时，必须要有输出项 C.在 C 语言中，整数可以以十二进制、八进制或十六进制地形式输出 D.调用 getchar 函数读入字符时，可以从键盘上输入字符所对应地 ASCII 码)

54、以下叙述中错误的选项是（）>(A.C 程序地基本组成单位是语句 B.C 程序中地每一行只能写一条语句 D.C 语句必须在一行内写完)

55、以下叙述中正确的是（）>(A.一个 C 源程序可由一个或多个函数组成 B.一个 C 源程序必须包含一个 main()函数 C.C 源程序的基本组成单位是函数)

56、以下叙述中正确的说法是（）>(A.C 程序中地#include 和#define 行均不是 C 语句 B.除逗号运算符外，赋值运算符地优先级最低 C.C 程序中，j++; 是赋值语句)

57、以下叙述中正确的选项是（）>(A.C 语言源程序经编译后生成后缀为 .obj 地目标程序 B.C 语言经过编译、连接步骤之后才能形成一个真正可执行地二进制机器指令文件 C.用 C 语言编写地程序称为源程序，它以 ASCII 代码形式存放在一个文本文件中)

58、以下选项中不合法的字符常量是（）>(A."B".C.68D.D)

59、以下选项中不是 C 语言合法整数的是（）>(B.386D.x2a2)

60、以下选项中不属于 C 语言的数据类型是（）>(A.复数型 B.逻辑型 D.集合型)

61、以下选项中能作为 C 语言合法常量地是（）>(B.100000C.aD.'011')

62、以下正确的说法是（）>(A.在不同函数中可以使用相同地名字地变量 B.形式参数是局部变量 C.在函数内定义地变量只在本函数范围内有效)

63、在 C 语言程序设计中，一段完整的程序一般包含哪些控制流程（）>(A.顺序类流程 B.分支类流程 D.跳转类流程)

64、在 C 语言中，不合法的长整型常数是（）>(A.0C.0. 054838743D.2186900000)

65、在 C 语言中，函数的数据类型是指（）>(B.函数形参的数据类型 C.调用该函数时的实参的数据类型 D.任意指定的数据类型)

66、在 C 语言中，引用数组元素时，其数组下标地数据类型不允许是（）>(A.整型常量 C.实型常量 D.任何类型的表达式)

67、在函数调用时，以下说法不正确的是（）>(A.函数调用后必须带返回值 C.函数间的数据传递不可以使用全局变量 D.主调函数和被调函数总是在同一个文件里)

68、在缓冲文件系统中用于读写的函数有（）>(A.putchar()B.fread()D.purw())

69、这段程序中的常量有:#definePI3. 14;voidmain（）{intsum;sum=10+15;printf（"sum=%d",sum）;printf（"resultis25"）;}这段程序中常量有（）>(A.10C.15D.PI)

70、指出下面错误的表达式（）>(B.i=i++D.56=a11)

71、指出下面合法的标识符（）>(A.3abB.ABCC.a 3)

72、指出下面哪些是合法的用户定义的标识符（）>(B. _908C.wind)

73、属于单字符运算符的是（）>(A. (B.+C.=D.))

判断(505)--电大资源网: <http://www.dda123.cn/> (微信搜: 905080280)

1、abs 函数的原型被包含在 include 子目录下的 stdlib.h 头文件和 math.h 头文件中。>对

2、break 语句带有语句体。>错

3、break 语句可以使用在 switch 语句中。>对

4、break 语句只能使用在循环语句中。>错

5、break 语句只有保留字，没有语句体。>对

6、continue 语句只有保留字，没有语句体。>对

7、continue 只能使用在任一种循环语句的循环体中。>对

8、C 语言把对文件输入输出的数据，当做一行行的文本来处理，才会出现换行时的自动转换现象，这种文件操作模式被称作文本模式。>对

9、C 语言系统中提供一个用来描述文件属性的类型标识符为 FILE，它与系统头文件 stdio.h 无关。>对

10、C 语言系统中提供一个用来描述文件属性的类型标识符为 FILE，它与系统头文件 stdio. h 无关。>错

11、C 语言系统中提供一个用来描述文件属性的类型标识符为 FILE。>对

12、C 语言中，数组的下标是从 0 开始的。>对

13、C 语言中，转义字符"\n"的功能是换行。>对

14、C 语言中的变量名只能由字母、数字和下划线三种字符组成，且第一个字符必须为字母或者下划线。>对

15、C 语言中的标准输入和输出设备文件都属于字符文件。>对

16、C 语言中的每条复合语句以花括号作为结束符。>错

17、C 语言中的每条简单语句以分号作为结束符。>对

18、C 语言中的数据文件包括字符文件和字节文件这两种不同的存储类型。>对

19、C 语言中的数据文件只包括字符文件这一种存储类型。>错

20、C 语言中的运算符有单目、双目和 3 目之分。>对

21、C 语言中的运算符只有单目和双目两种，没有 3 目运算符。>错

22、do-while 循环是先判断循环条件，当条件为真时执行循环体。>错

23、do-while 循环是先执行循环体，然后进行循环条件判断。>对

24、do-while 循环语句的循环体，可以是简单语句，也可以是复合语句。>对

25、do-while 循环语句的循环体，只能是复合语句，不能是简单语句。>错

26、do-while 循环语句是以分号结束的语句。>对

27、do-while 循环语句是以循环体结尾的。>错

28、do-while 循环语句中的循环体可能不会被执行。>错

29、do-while 循环语句中的循环体至少被执行一次。>对

30、float 的类型长度为 8。>错

31、for 循环是先判断循环条件，当条件为真时执行循环体。>对

32、for 循环语句不能够被转换为 while 循环语句。>错

33、for 循环语句的头部，for 后面圆括号内共有个表达式位置。>错

34、for 循环语句的循环体，可以是简单语句，也可以是复合语句。>对

35、for 循环语句能够被转换为 while 循环语句。>对

36、for 循环语句中的循环体可能不会被执行。>对

37、for 循环语句中的循环体至少被执行一次。>错

38、NULL 是一个符号常量，通常作为空指针值，它代表的值为 0。>对

39、return 语句只有保留字，没有其他成分。>错

40、return 语句中可以带有返回表达式。>对

41、strcmp 函数用于进行两个字符串之间的比较。>对

42、strcpy 函数用于把一个字符串拷贝到另一个字符数组空间中。>对

43、while 循环是先判断循环条件，当条件为真时执行循环体。>对

44、while 循环是先执行循环体，然后进行循环条件判断。>错

45、while 循环语句的循环体，只能是复合语句，不能是简单语句。>错

46、while 循环语句中的循环体可能不会被执行。>错

47、while 循环语句中的循环体至少被执行一次。>错

48、while 语句又称为 while 循环，是一种循环性语句，它的循环体是一条语句。>错

49、表达式 1%5 的值为 5。>错

50、表达式 14%5 的值为 5。>错

51、表达式 21%3 的值为 0。>对

52、表达式 45%13 的值为 3。>错

53、表达式 sizeof (25) 的值等于 4。>对

54、表达式 x=x+1 表示成增量表达式为 ++x。>对

55、表达式 x=x+1 表示成增量表达式为 x++。>错

56、表达式 x=x+y 表示成复合赋值表达式为 x+=y。>对

57、表达式 x=x+y 表示成复合赋值表达式为 x+=y。>错

58、表达式 x=x-1 表示成减量表达式为 --x。>对

59、表达式 x=x-1 表示成减量表达式为 x--。>错

60、表达式 (float) 25/4 的值为 6. 25。>对

61、表达式 (float) 25/4 的值为 6。>错

62、表达式 (int) 14.6%5 的值为 4。对

63、表达式 (int) 14. 6%5 的值为 0。>错

64、不能够在定义结构类型的同时定义它的变量。>错

65、常数-7. 00 是双精度定点数。>对

66、常数 3. 26e4 是双精度定点数。>错

67、常数 3. 26f 是单精度浮点数。>错
 68、常数 3. 26 是双精度定点数。>对
 69、初始化结构体数组有两种方式：先定义结构体数组类型再初始化结构体数组、在定义结构体数组的同时初始化结构体数组。>对
 70、从文本文件中输入数据时，需要调用系统函数 fputc 或 fgets。>错
 71、存储包含一个字符的字符串需要占用 1 个字节的存储空间。>错
 72、存储包含一个字符的字符串需要至少占用 2 个字节的存储空间。>对
 73、存储一个长度为 0 的空字符串至少需要占用 1 个字节的存储空间。>对
 74、存储一个空字符串需要占用 0 个字节的存储空间。>错
 75、存储字符'a'需要占用 1 个字节的存储空间。>对
 76、打开一个文件，对该文件进行访问操作之后，是否关闭该文件是可有可无的。>错
 77、带有随机函数的表达式 rand%20 的值所在的区间范围是 0~19。>对
 78、当不需要函数返回任何值时，则需要使用保留字 void 作为函数的返回值类型。>对
 79、当从字符文件读取到回车和换行这两个连续字符时，将忽略回车符。>对
 80、当向字符文件输出一个换行符时，实际将输出的是回车符。>错
 81、当向字符文件输出一个换行符时，实际将输出的是回车符或换行符。>错
 82、当向字符文件输出一个换行符时，实际将输出的是回车和换行这两个字符。>对
 83、当需要在程序文件中引入系统头文件时，使用的是#include 命令。>对
 84、当需要在程序文件中引入用户定义的头文件时，使用的不是#include 命令。>错
 85、当循环次数确定时，while 循环语句不能转换为 for 循环语句。X
 86、当循环次数确定时，while 循环语句能够转换为 for 循环语句。>对
 87、调用系统函数时，要先使用#include 命令包含该系统函数的原型语句所在的系统头文件。>对
 88、调用字符串函数 strcat 时需要带有一个参数。>错
 89、调用字符串函数 strcmp 时需要带有两个参数。>对
 90、调用字符串函数 strcpy 时需要带有一个参数。>错
 91、调用字符串函数 strlen 时需要带有一个参数。>对
 92、定点数包括单精度定点数和双精度定点数两种。>对
 93、定点数既有双精度表示，又有单精度表示。对
 94、定点数只有双精度表示，没有单精度表示。>错
 95、定义符号常量有两种方法，一种是采用定义语句的方法，另一种是采用预处理命令的方法。>对
 96、定义符号常量只有一种方法，就是采用常量定义语句的方法。>错

97、定义符号常量只有一种方法，就是采用预处理命令#define 的方法。>错
 98、定义结构类型的变量时，不能够同时进行初始化。>错
 99、定义结构类型的变量时，能够同时进行初始化。>对
 100、定义结构类型的变量时，能够用同类型的其他变量进行初始化，不能用数据表进行初始。>错
 101、定义结构类型的变量时，能够用同类型的其他变量进行初始化。>对
 102、定义结构类型的同时能够定义它的变量。>对
 103、定义结构体类型变量 teach1 的语句为 struct{intnum;intage;}teacher; structteacherteach1; >错
 104、动态存储分配具有这两个特点，第一是不需要预先分配存储空间，第二是根据程序需要即时分配，也可以根据程序的需要扩大或缩小。>对
 105、短整型、标准整型和长整型的长度分别为 2、4 和 8。>错
 106、对二进制文件进行读和写操作将使用不同的系统函数。>对
 107、对二进制文件进行读和写操作将使用同一个系统函数。>错
 108、对二进制文件进行读和写操作将使用相同的系统函数。>错
 109、对于标识为 static 属性的变量，若没有对它进行初始化，则默认的初值为 0。>对
 110、对于不加 static 标识的局部变量，若没有对它进行初始化，则它的初值是不确定的。>对
 111、对于全局或文件作用域变量，其定义位置可以在任何一个函数定义之内。>错
 112、对于全局或文件作用域变量，其定义位置只能在程序文件中所有函数定义之外。>错
 113、对于全局或文件作用域变量，若没有对它进行初始化，则默认的初值为 0。>对
 114、对于一个长度为 n 的字符串，存储它的字符数组的长度至少为 n。>错
 115、对于一维数组元素，数组名后使用一对中括号。>对
 116、多行注释语句的开始和结束标记符可以互换使用。>错
 117、二维数组可以看成是由多个一维数组构成的一维数组。>错
 118、二维数组元素带有一个下标。>错
 119、访问结构成员访问运算符只有一种方式，即使用点运算符。错
 120、访问一个链表时，必须从表头指针出发，依次访问每个结点。>对
 121、访问一个链表时，不必从表头指针出发，可以随时访问任一结点。>错
 122、浮点数包括单精度浮点数和双精度浮点数两种。>对
 123、浮点数只有单精度表示，没有双精度表示。>错
 124、关系表达式 x+y>5 的相反表达式为 x+y<=5>对
 125、关系表达式 (x!=0) 的等价表达式为 (!x)。>错
 126、关系表达式 (x!=0) 的等价表达式为 (x)。>对
 127、关系表达式 (x==0) 的等价表达式为 (!x)。>对
 128、广义整数类型包括短整型、标准整型和长整型 3 种。>对
 129、函数被调用执行中，形变量的初值由调用该函数时对应的实参值给定。>对
 130、函数调用表达式 exit (1) 和 srand (10) 都是无值表达式。>对

131、函数调用参数表中的每个实参可以是表达式。>对
 132、函数调用参数表中的每个实参只能是常量或变量。>错
 133、函数调用格式中的参数表被称为实参表。>对
 134、函数调用格式中的参数表被称为形参表。>错
 135、函数定义格式中的参数表被称为形参表。>对
 136、函数形参变量不属于局部变量。>错
 137、计算机系统将为每个数据分配一定大小的存储空间，此存储空间称为一个数据单元。>对
 138、假定 a 是一个二维数组，则 a+i 的值与&a[i]的值不等。>错
 139、假定 a 是一个二维数组，则 a+i 的值与&a[i]的值相等。>对
 140、假定 a 是一个二维数组，则表达式*(a[j]+i) 所对应的元素为 a[i][j]。>错
 141、假定 a 是一个二维数组，则进行 a++运算是不允许的。>对
 142、假定 a 是一个二维数组，则进行 a++运算是允许的。>错
 143、假定 a 是一个二维数组，则数组名 a 的值和&a[0]的值不等。>错
 144、假定 a 是一个二维数组，则数组名 a 的值和 a[0]的值相等。>对
 145、假定 a 是一个一维数组，则 a+i 的值与&a[i]的值相等。>对
 146、假定 a 是一个一维数组，则 a+i 的值与 a[i]的值相等。>错
 147、假定 a 是一个一维数组，则 a+i 的值与&a[i]的值相等。>对
 148、假定 a 是一个一维数组，则表达式*(a+i) 所对应的元素为 a[i]。>对
 149、假定 a 是一个一维数组，则表达式*(a+i) 所对应的元素为 a[i - 1]。>错
 150、假定 a 是一个一维数组，则进行 a++运算是不允许的。>对
 151、假定 a 是一个一维数组，则进行 a++运算是允许的。>错
 152、假定 a 是一个一维数组，则数组名 a 的值和&a[0]的值相等。>对
 153、假定 a 是一个指针数组，则 a+i 所指对象的地址比 a 地址大 4*i 个字节。>对
 154、假定 a 是一个指针数组，则 a+i 所指对象的地址比 a 地址大 4*i 字节。>对
 155、假定 a 是一个指针数组，则 a+i 所指对象的地址比 a 地址大 i 个字节。>错
 156、假定 char 类型变量占用 1 个字节,且数组定义如下,则数组 tab_str 在内存中所占字节数是 810。chartab_str[10][81]; >对
 157、假定 p 所指对象的值为 25，p+1 所指对象的值为 42，则表达式*++p 的值为 42。>错
 158、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则*++p 的值为 25。>错
 159、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则*p++ 的值为 46。>错
 160、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则执行*(p++) 运算后，p 所指对象的值为 46。>对
 161、假定 p 所指对象的值为 25，p-1 所指对象的值为 42，则执行*p--运算后，p 所指对象的值为 42。>错
 162、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则执行*(p++) 运算后，p 所指对象的值为 46。>对
 163、假定 x=5，则表达式 2*x++ 的值为 12。>错
 164、假定 x=5，则表达式 2+x++ 的值为 7。>对

165、假定 $x=5$ ，则执行“ $a=(x?10:20)$ ；”语句后 a 的值为 10。>对

166、假定 $x=5$ ，则执行“ $y=++x$ ；”语句后， y 的值为 6。>对

167、假定 $x=5$ ，则执行“ $y=x++B$ 。”语句后， x 的值为 5。>错

168、假定 x 是一个逻辑量，对于 x 的任何值，表达式 $(x!x)$ 的值始终为假。>错

169、假定 x 是一个逻辑量，对于 x 的任何值，表达式 $(x\&\&!x)$ 的值始终为真。>错

170、假定 x 为一个简单变量，则 $\&x$ 表示 x 的地址。>对

171、假定 $y=10$ ，则表达式 $++y*3$ 的值为 30。>错

172、假定二维数组的定义为“ $\text{chara}[M][N]B$ ”，则该数组所含元素的个数为 $M+N$ 。>错

173、假定二维数组的定义为“ $\text{doublea}[M][N]$ ；”，则每个数组元素的列下标取值范围在 $0\sim N$ 之间。>错

174、假定二维数组的定义为“ $\text{doublea}[M][N]$ ；”，则每个数组元素的行下标取值范围在 $0\sim M-1$ 之间。>对

175、假定二维数组的定义为“ $\text{inta}[3][5]B$ ”，则该数组所占存储空间的字节数为 60。>对

176、假定二维数组的定义语句为“ $\text{doublea}[M][N]$ ；”，则每个数组元素的列下标取值范围在 $0\sim N-1$ 之间。>错

177、假定二维数组的定义语句为“ $\text{doublea}[M][N]$ ；”，则每个数组元素的行下标取值范围在 $0\sim M-1$ 之间。>对

178、假定是一个一维数组，则表达式 $*Ca+i$ 所对应的元素为 $[]$ 。>对

179、假定是一个一维数组，则进行 $a++$ 运算是不允许的。>对

180、假定要访问一个结构对象 x 中的由 a 指针成员所指的对象，则表示方法为 $x.a$ 。>错

181、假定要访问一个结构指针 p 所指对象中的 b 指针成员所指的对象，则表示方法为 $p->b$ 。>错

182、假定一个结构类型的定义为“ $\text{structA}(\text{inta},\text{bBA}*\text{cB})B$ ”，则该类型的长度为 12。>对

183、假定一个结构类型的定义为“ $\text{structB}(\text{inta}[5]B\text{char}*\text{bB})B$ ”，则该类型的长度为 20。>错

184、假定一个枚举类型的定义为“ $\text{enumRB}\{\text{ab},\text{ac}=3,\text{ad},\text{ae}\}x=\text{ad}$ ；”，则 x 的值为 2。>错

185、假定一个数据对象为 inH 类型，则指向该对象的指针类型为 irl 仲势。>对

186、假定一个数据对象为 $\text{int}*$ 类型，则指向该对象的指针类型仍为 $\text{int}*$ 。>错

187、假定一个数据对象为 $\text{int}*$ 类型，则指向该对象的指针类型仍为 $\text{int}*$ 类型。>错

188、假定一个数据对象为 $\text{int}*$ 类型，则指向该对象的指针类型为 $\text{int}**$ 。>对

189、假定一个数据对象为 x ，则 x 的地址表示为 $*x$ 。>错

190、假定一维数组的定义为“ $\text{char}*\text{a}[8]$ ；”，则数组 a 中元素的类型为字符指针型。>对

191、假定一维数组的定义为“ $\text{char}*\text{a}[8]B$ ”，则该数组所含元素的个数大于 8。>错

192、假定一维数组的定义为“ $\text{doublea}[M]$ ；”，则每个数组元素的下标取值范围在 $1\sim M$ 之间。>错

193、假定一维数组的定义为“ $\text{char}*\text{a}[8]$ ；”，则数组 a 中元素的类型为字符型。>错

194、假定一维数组的定义为“ $\text{char}*\text{a}[8]$ ；”，则数组 a 中元素的类型为字符指针型。>对

195、假定一维数组的定义为“ $\text{doublea}[M]$ ；”，则每个数组元素的下标取值范围在 $0\sim M-1$ 之间。>对

196、减量表达式 $--y$ 表示成赋值表达式为 $y=y-1$ 。>对

197、结构成员访问运算符有两个，点运算符（.）和箭头运算符（->）。>对

198、结构成员访问运算符有三个，其中两个为点运算符（.）和箭头运算符（->）。>错

199、进行动态存储分配的 calloc 函数带有两个参数。>对

200、进行动态存储分配的 calloc 函数带有两个参数。>对

201、进行动态存储分配的 calloc 函数带有一个参数。>对

202、进行动态存储分配的 malloc 函数带有两个参数。>错

203、进行动态存储分配的 malloc 函数带有一个参数。>对

204、进行动态存储分配的 malloc 函数带有一个参数。>错

205、可以利用“递归”进行具有自重复性无限重复动作的执行，即“递归计算”或“递归执行”。>错

206、空类型是 C 语言的一种数据类型。>对

207、空字符串的长度为 1。>错

208、逻辑表达式 $(a\&\&b\&\&b==5)$ 的相反表达式为 $(a\&\&b\&\&b==5)$ 。>错

209、逻辑表达式 $(a\>b\&\&b==5)$ 的相反表达式为 $(a\>b\&\&b==5)$ 。>错

210、逻辑表达式 $(x\&\&3\&\&x\&\&10)$ 的相反表达式为 $(x\&\&3\&\&x\&\&10)$ 。>错

211、逻辑表达式 $(x\>3\&\&x\<10)$ 的相反表达式为 $(x\<=3\&\&x\>=10)$ 。>对

212、每次只能从文本文件中读取一个字符，不能一次读取一行字符串。>错

213、每个 C 程序文件在编译时可能出现有警告性误，其应的标识符为 error 。>错

214、如果需要从被调用函数返回一个函数值，被调用函数必须包含 return 语句。>对

215、如果一个函数只允许同一程序文件中的函数调用，则不应在该函数定义的开始前加上保留字 static 。>错

216、如果在一个函数体中又出现对自身函数的调用，此种函数调用被称为递归调用。>对

217、若 a 和 b 均是整形变量并已正确赋值，正确的 switch 语句是 $\text{switch}a+b*$ 。>错

218、若 p 指向 x ，则 $*p$ 与 x 的值不同。>错

219、若 $x=15$ ， $y=10$ ，则 $x!=y$ 的逻辑值为真。>对

220、若 $x=15$ ， $y=10$ ，则 $x==y$ 的逻辑值为真。>错

221、若 $x=15$ ， $y=20$ ，则 $x>y$ 的逻辑值为真。>错

222、若 $x=5$ ， $y=10$ ，则 $x!=y$ 的逻辑值为真。>对

223、若 $x=5$ ， $y=10$ ，则 $x>y$ 逻辑值为真。>错

224、若 $x=5$ ， $y=10$ ，则 $x\<=y$ 的逻辑值为假。>错

225、若 $x=5$ ， $y=10$ ，则 $x\<=y$ 的逻辑值为真。>对

226、若 $x=5$ ， $y=8$ ，则 $x\<=y$ 的逻辑值为真。>对

227、若对一个文本文件以“ r ”方式打开时，则称该文本文件为输出文件。>错

228、若需要定义一个符号常量，并且使 C 语言能够进行类型检查，则应在定义语句的开始使用保留字 const 。>对

229、若要把一个整型指针 p 转换为字符指针，则采用的强制转换表达式为 $(\text{char}*)p$ 。>对

230、若要在程序文件中使用标准输入和输出函数，则需要引入的系统头文件为 stdio.h 。>对

231、若要在程序文件中使用数学函数，则需要引入的系统头文件为 math.h 。>对

232、若用数组名作为函数调用时的实参，则实际上传递给形参的是数组首地址。>对

233、若有定义： $\text{doublex}[3][5]$ ；则数组中列下标的上限为 4。>对

234、设 $a=3$ ， $b=2$ ， $c=1$ ，则表达式 $f=a>b>c$ 的值是 0。>对

235、设 a 为 int 型变量，执行下列赋值语句 $a=125.534$ ；后， a 的取值是 125。>对

236、设 $\text{inta}=12$ ，则执行完语句 $a-=a*a$ 后， a 的值是 0。>对

237、设 int 型变量 x 的值为 9，那么表达式 $x--x--x--$ 的值为 27。>错

238、十进制数 25 表示成符合 C 语言规则的八进制数为 31。>错

239、十进制数 25 表示成符合 C 语言规则的十六进制数为 $0x19$ 。>对

240、实参是在调用时传递给函数的参数，即传递给被调用函数的值。>对

241、使用 $\#include$ 命令只能包含头文件，不能包含其他任何文件。>错

242、使用“ $\text{typedefcharBB}[10]$ ；”语句定义标识符 BB 为含有 10 个元素的一维字符数组。>错

243、使用“ $\text{typedefcharBB}[10][50]$ ；”语句定义标识符 BB 为含有 10 行 50 列共 500 个元素的二维字符数组。>对

244、使用“ $\text{typedefcharBB}[10][50]B$ ”语句定义标识符 BB 为含有 10 行 50 列的二维字符数组类型。>对

245、使用 const 语句定义一个符号常量时，不必它进行初始化。>错

246、使用 const 语句定义一个符号常量时，不需要对它同时进行初始化。>错

247、使用 const 语句定义一个符号常量时，则必须对它同时进行初始化。>对

248、使用 sizeof 运算符能够求出一种数据类型的长度。>对

249、使用“ $\text{typedefcharBB}[10][50]$ ；”语句定义标识符 BB 为含有 10 行 50 列共 500 个元素的二维字符数组。>错

250、使用“ $\text{typedefcharBB}[10]$ ；”语句定义标识符 BB 为含有 10 个元素的一维字符数组。>错

251、使用“ $\text{typedefcharBB}[10]$ ；”语句定义标识符 BB 为含有 10 个元素的一维字符数组类型。>对

252、使用“ $\text{typedefintCC}[4][6]$ ；”语句之后，再使用“ CCa ；”语句，则数组 a 所占用的存储空间的字节数为 96。>对

253、使用结构类型时，必须一起使用关键字 struct 和类型标识符。>对

254、使用结构类型时，只需要使用类型标识符，不需要使用关键字 struct。>错

255、使用三重嵌套循环来访问或修改元素的值。>对

256、使用一个结构类型时，必须一起使用关键字 struct 和类型标识符。>对

257、为了存储一个长度为 n 的字符串，所使用的字符数组的长度至少为 n+1。>对

258、为了结束本层循环类语句或 switch 语句的执行过程，在语句体中需要使用 break 语句。>对

259、系统默认的 C 语言源程序文件的扩展名是.cpp。>错

260、向文本文件中输出数据时，需要调用系统函数 fputc 或 fputs。>错

261、循环类语句包括 if 语句、while 语句和 do-while 语句三种。>错

262、一个 C 程序可以有多个 main 函数。>错

263、一个 C 语言程序能够包含多个用户头文件。>对

264、一个 C 语言程序只能包含一个程序源文件。>错

265、一个 C 语言程序只能包含一个用户头文件。>错

266、一个磁盘数据文件的文件名由文件主名和扩展名所组成，其中间用圆点分开。>对

267、一个二维数组名是一个指针常量，其值可以被修改。>错

268、一个二维字符数组 a[10][20]能够存储的每个字符串的长度不超过 20。>错

269、一个二维字符数组 a[10][20]能够最多存储 9 个字符串。>错

270、一个二维字符数组 a[10][20]中存储每个字符串的最大长度为 19。>对

271、一个二维字符数组 a[10][20]中存储每个字符串的最大长度为 20。>错

272、一个函数必须返回一个值，不允许定义不返回任何值的函数。>错

273、一个结构类型的长度等于所含的所有数据成员类型的长度之和。>对

274、一个数组名是一个指针常量，其值可以被修改。>错

275、一个数组名是一个指针常量，其值不允许被修改。>对

276、一个数组名是一个指针常量，其值可以被修改。>错

277、一个指针指向一个数据对象时，指针值是这个对象的首字节地址。>对

278、一条注释语句的最后必须使用分号结束。>错

279、已知“inta[10],*p=a;”，则进行*p++运算是允许的。>错

280、已知“inta[3][5];int (*p)[5]=a;”，则进行 p++运算是允许的。>错

281、已知 x=-25，则 abs(x) 的值为 25。>对

282、已知 x=-25，则 fabs(x) 的值为-25.0。>错

283、已知“inta[10],*p=a;”，则进行 p++运算是允许的。>错

284、已知“inta[10],*p=a;”，则进行 p++运算是允许的。>对

285、已知“inta[3][5];int (*p)[5]=a;”，则进行 p++运算是允许的。>对

286、已知函数定义为：intstat(inta, float*p)，则此函数的 2 个参数类型分别是整型、指针型。>对

287、已知字符'C'的 ASCII 码为 67，当执行“intx='C'+5”>对

288、已知字符'C'的 ASCII 码为 67，当执行“intx='C'+5;”语句后 x 的值为 72。>对

289、用 C 语言编写出一个完整的程序后，第一步需要上机建立相应的工作区和项目并建立、输入和编辑该程序中的相应文件，其中有并且只有一个程序文件必含有一个并且只有一个主函数。>对

290、用户使用类型定义语句能够定义出除标准类型之外的用户所需要的数据类型。>对

291、用于释放动态存储空间的函数为 free，调研它需要带有一个参数。>对

292、用于释放动态存储空间的函数为 free，调用它不需要带有任何参数。>错

293、用于释放动态存储空间的函数为 free，调用它需要带有一个参数。>对

294、用于输出表达式值的标准输出函数是 printf。>对

295、有如下函数头：intmax(inti, intj)则此函数有 2 个参数。>对

296、有一个语句为“chars2[5]={"abcd"};”，s2 数组长度是合适的。>对

297、有一条语句为“chars2[4]={"abcd"};”，s2 数组长度的定义是合适的。>错

298、与结构成员访问表达式 (*fp).score 等价的表达式是 fp->score。>对

299、语言程序中，对于递归函数和非递归函数，其函数头部有区别标志。>错

300、语言程序中，只有一种使用注释语句的方法。>错

301、语言程序中，主函数也可以成为递归函数。>错

302、语言中，在定义一个字符数组时，不能利用一个字符串进行初始化。>错

303、源程序文件被编译成目标文件后，其目标文件中也存在对应的注释内容。>错

304、源程序文件被编译成目标文件后，源程序文件中的注释语句仍然存在。>对

305、在“printf(“%d-%d=%d”,x,y,x-y);”语句所使用的格式字符串中，包含有 3 个格式符。>错

306、在“printf(“s=%c”,“abc”);”语句中，格式符的使用同后面的输出数据相适合。>错

307、在“scanf(“%d%d”,&x,&y);”语句所使用的格式字符串中，包含有 2 个格式符。>对

308、在“scanf(“%d%d”,&x,&y);”语句中，x 和 y 变量均为整数类型。>对

309、在“scanf(“%d-%d=%d”,&x,&y,&z);”语句中，有语法错误。>错

310、在“scanf(“%s”,x);”语句中，x 变量不是字符指针类型。>错

311、在 C 语言程序上机操作的过程中，编译程序的操作步骤之后是运行操作。>错

312、在 C 语言程序上机操作的过程中，编译程序的操作步骤之前是输入和编辑程序。>对

313、在 C 语言程序上机操作的过程中，需要对每个用户头文件进行单独的编译操作。>错

314、在 C 语言程序中，ABC 和 abc 被作为同一标识符使用。>错

315、在 C 语言程序中，RBC 和 Rbe 可以被作为同一个标识符使用。>错

316、在 C 语言程序中，除主函数外的任何用户定义的函数都可以是递归函数。>对

317、在 C 语言程序中，存储字符串的方法是使用字符数组。>对

318、在 C 语言程序中，存储字符串的方法是使用字符指针。>错

319、在 C 语言程序中，对于递归函数和非递归函数，其函数头部无区别标志。>对

320、在 C 语言程序中，对于递归函数和非递归函数，其函数头部有区别标志。>错

321、在 C 语言程序中，有两种使用注释语句的方法。>对

322、在 C 语言程序中，在行尾使用注释的开始标记符为一单斜线字符。>对

323、在 C 语言程序中，只有一种使用注释语句的方法。>错

324、在 C 语言程序中，主函数可以是递归函数。>错

325、在 C 语言程序中，主函数也可以成为递归函数。>错

326、在 C 语言中，保留字是有专门含义和作用的，不能作为一般标识符使用。>对

327、在 C 语言中，常数 28 和 326 具有相同的数据类型。>错

328、在 C 语言中，打开一个数据文件的系统函数为 fopen，它带有两个字符串参数，用来给定文件名和打开方式。>对

329、在 C 语言中，打开一个数据文件的系统函数为 fopen，它带有一个字符串参数，用来给定待操作的文件名。>错

330、在 C 语言中，对二进制文件的所有不同打开方式共有 6 种。>对

331、在 C 语言中，逻辑运算符“||”称为逻辑或。>对

332、在 C 语言中，每个保留字是有专门含义和作用的，不能作为一般标识符使用。>对

333、在 C 语言中，求一个实数的绝对值的数学函数为 fabs(x)。>对

334、在 C 语言中，求一个整数的绝对值的数学函数为 abs(x)。>对

335、在 C 语言中，退出程序运行的系统函数为 exit。>对

336、在 C 语言中，系统函数 fseek 用来移动数据文件中的文件位置指针，它仅带有一个文件流参数。>错

337、在 C 语言中，系统函数 fseek 用来移动数据文件中的文件位置指针。>对

338、在 C 语言中，系统函数 fseek 用来移动数据文件中的文件位置指针。>对

339、在 C 语言中，一个函数由函数头和函数体组成。>对

340、在 C 语言中，一个结构类型的长度等于所含的所有数据成员类型长度的最大值。>错

341、在 C 语言中，一个结构类型的长度等于所含的所有数据成员类型的长度之和。>对

342、在 C 语言中，以'\n'作为字符串结束标志。>错

343、在 C 语言中，在定义一个指针变量时，不能对其进行初始化。>错

344、在 C 语言中，在定义一个字符数组时，不能利用一个字符串进行初始化。>错

345、在 C 语言中，在定义一个字符数组时，可以同时利用一个字符串进行其初始化。>对

346、在 C 语言中,正弦、余弦或正切函数的参数的度量单位是度。>错

347、在 C 语言中,转义字符'\n'表示一个换行符。>对

348、在 C 语言中,作为标识符的第一个字符只能是英文字母或下划线,不能是数字字符。>对

349、在 C 语言字符集中,包含有全部 26 个英文大写字母和对应的小写字母。>对

350、在 C 语言字符集中,一个大写英文字母和它的小写英文字母被视为不同的字符。>对

351、在 for 循环语句的头部,for 后面括号内的各表达式之间使用逗号分开。>错

352、在 for 循环语句的头部,for 后面括号内共有 2 个表达式位置。>错

353、在 for 循环语句的头部,for 后面圆括号内共有 3 个表达式位置。>对

354、在 if 语句中,必须带有 else 子句,不能省略它。>错

355、在 if 语句中,可以带有 else 子句,也可以省略它。>对

356、在 if 语句中,每个 else 关键字与它前面同层次并且最接近的 if 关键字相配套。>对

357、在 switch 语句中,每个 case 和冒号之间的表达式只能是常量。>对

358、在“printf(“%d-%d=%d”,x,y,x-y);”语句所使用的格式字符串中,包含有 3 个格式符。>对

359、在“printf(“s=%c/n”,“abc”);”语句中,格式符的使用同后面的输出数据相适合。>错

360、在“scanf(“%d%d”,&x,&y);”语句所使用的格式字符串中,包含有 2 个格式符。>对

361、在“scanf(“%d%d”,&x,&y);”语句中,x 和 y 变量均为整数类型。>对

362、在“scanf(“%d-%d=%d”,&x,&y,&z);”语句中,有语法错误。>错

363、在“scanf(“%s”,x);”语句中,x 变量不是字符指针类型。>错

364、在变量定义语句中,被定义的变量必然出现在赋值号的左边。>对

365、在程序文件中添加注释语句是为了增强程序的可读性。>对

366、在程序文件中要进行动态存储分配和释放,则需要引入的系统头文件是 stdio. h。>错

367、在程序执行完成一个函数调用后,将结束整个程序的执行过程,返回到操作系统或 C 语言集成开发环境界面窗口。>错

368、在程序中调用系统函数时,必须使用#include 命令包含该系统函数所属的系统头文件。>对

369、在程序中执行到 break 语句时,将结束所在函数的执行过程,返回到调用该函数的位置。>错

370、在单行或多行注释中,其在释内容前需要使用标记符“/*”,而在结束处不需要使用“*/”。>错

371、在单行注释中,其注释内容前需要使用标记符“/*”,而在结束不需要使用“*/”。>错

372、在单行注释中,其注释内容前需要使用标记符“/”,而在结束时不需要使用任何标记符。>对

373、在递归函数的每次递归调用中,参数表得到的值不完全相同。>对

374、在递归函数的每次递归调用中,对于参数表中的每个形参,得到的值都相同。>错

375、在定义一个变量时,不能给它赋初值。>错

376、在定义一个变量时,不能够同时给它赋初值。>错

377、在定义一个变量时,可以同时给它赋初值。>对

378、在定义一个字符数组时,可以同时利用一个字符串进行其初始化。>对

379、在定义语句“intx=3,*p=&x,**pp=&p;”中的语法格式是正确的。>对

380、在定义语句“intx=3,*p=&x,**pp=p;”中的语法格式是正确的。>错

381、在定义语句序列“intx=3;constint*p=&x”中,定义 p 为整型常量指针。>对

382、在定义语句序列“intx=3; constint*p=&x;”中,p 所指对象为 x,并且可以通过 p 修改 x 的值。>错

383、在定义语句序列“intx=3; constint*p=&x;”中,定义 p 为整型常量指针。>对

384、在定义指针变量的语句“int*p,**pp;”中,p 和 pp 具有相同的指针类型。>错

385、在定义指针变量的语句“int*p,pp;”中,p 和 pp 具有不同的数据类型。>对

386、在定义指针变量的语句“int*p,**pp;”中,p 和 pp 具有相同的指针类型。>错

387、在定义指针变量的语句“int*p,pp;”中,p 和 pp 具有不同的数据类型。>对

388、在定义指针变量的语句“int*ppp;”中,p 和 pp 具有相同的数据类型。>错

389、在定义指针变量的语句“int*p, **pp;”中,p 和 pp 具有不同的指针类型。>对

390、在定义指针变量的语句中,对指针变量不允许赋初值。>错

391、在定义指针变量的语句中,对指针变量可以赋初值。>对

392、在定义指针变量的语句中,最前面的类型关键字不允许是指针类型。>错

393、在定义指针变量的语句中,最前面的类型关键字仍然允许是指针类型。>对

394、在二维数组的定义语句中,不能给数组中每个元素赋初值。>错

395、在二维数组的定义语句中,可以给数组中每个元素赋初值。>对

396、在二维数组的定义语句中,数组名后带有两对圆括号。>错

397、在二维数组的定义语句中,数组名后带有两对中括号。>对

398、在二维数组的定义语句中,数组名前为元素类型关键字。>对

399、在函数定义中,存在着调用自身函数的函数调用表达式,则称此函数为递归函数。>对

400、在函数定义中,存在着调用自身函数的函数调用表达式,则称此函数为非递归函数。>错

401、在函数模块之外定义的变量称为全局变量,若没有被初始化则系统隐含对它赋初值 0。>对

402、在函数体中定义一个变量时,若不选用任何存储属性关键字进行标识,则它为静态变量。>错

403、在结构成员访问运算符中,点运算符(.)为双目运算符。>对

404、在结构成员访问运算符中,点运算符(.)为双目运算符。>对

405、在结构成员访问运算符中,点运算符和箭头运算符的左边均为结构变量。>错

406、在结构成员访问运算符中,点运算符和箭头运算符的左边均为结构指针变量。>错

407、在结构成员访问运算符中,箭头运算符(->)为单目运算符。>错

408、在结构类型的定义中,不允许出现结构类型不完整的超前定义。>错

409、在结构类型的定义中,结构类型的作用域范围,与它的定义位置有关。>对

410、在结构类型的定义中,结构类型名在关键字 struct 和定义体之间。>对

411、在结构类型的定义中,其中的数据成员可以是本身类型的直接对象。>错

412、在结构类型的定义中,允许出现结构类型不完整的超前定义。>对

413、在利用 printf 函数向屏幕输出数据时,不需要使用格式字符串来规定输出数据的类型。>错

414、在利用 printf 函数向屏幕输出数据时,需要在参数表的开始位置使用相应的格式字符串。>对

415、在利用 scanf 函数从键盘输入数据时,不需要使用格式字符串来规定输入数据的类型。>错

416、在利用 scanf 函数从键盘输入数据时,需要在参数表的开始位置使用相应的格式字符串。>对

417、在数据文件打开方式字符串中,字符 r、w 和 a 具有确定的含义,分别代表读、写和追加方式。>对

418、在数据文件打开方式字符串中,字符 r 和 w 具有确定的含义,r 代表写文件方式,w 代表读文件方式。>错

419、在数据文件打开方式字符串中,字符加号(+)具有确定的含义,代表既允许读也允许写的打开方式。>对

420、在数据文件打开方式字符串中,字符加号(+)只用于打开二进制文件的方式中,不能用于打开文本文件的方式中。>错

421、在同一个结构类型的定义中,各数据成员名必须不同。>对

422、在同一个结构类型的定义中,各数据成员名允许相同。>错

423、在同一个作用域不可定义同名变量,在不同的作用域可以定义同名变量。>对

424、在一个 C 语言程序文件中,若要包含一个头文件,则使用以百分号开始的预处理命令。>错

425、在一个 C 语言程序中,命名为 main 的主函数只能有一个,不允许出现多个。>对

426、在一个 C 语言程序中,主函数的函数名可以不是标识符 main。>错

427、在一个程序文件中，允许出现结构类型不完整的超前定义。>对

428、在一个磁盘数据文件的文件名中，文件主名和扩展名都是必须的，不可省略。>错

429、在一个磁盘数据文件的文件名中，文件主名是必须的，扩展名可任选使用。>对

430、在一个函数定义中，参数表不能为空。>错

431、在一个函数定义中，参数表可以为空。>对

432、在一个函数定义中，参数表中的参数定义之间是采用分号分开的。>错

433、在一个函数定义中，参数表中的参数类型不能为 void 类型。>对

434、在一个函数定义中，参数表中的参数类型可以为 void 类型。>错

435、在一个函数定义中，参数表中的参数说明项之间是采用逗号分开的。>对

436、在一个函数定义中，参数表中的参数说明项之间是采用分号分开的。>错

437、在一个函数定义中，函数体的格式是一条复合语句。>对

438、在一个函数定义中，函数体的格式是一条复合语句。假定 a 是一个一维数组，则表达式*(a+i)所对应的元素为 a[i]。>错

439、在一个函数定义中，函数体是一条复合语句。>对

440、在一个函数定义中，函数体是一条简单语句。>错

441、在一个函数定义中，函数体是用一对花括号括起来的。>对

442、在一个链表的结点结构中，必然包含有一个指向自身结点类型的指针域。>对

443、在一个链表的结点结构中，存在着一个指针域，但不一定指向自身结点类型。>错

444、在一个链表中，表头结点和表尾结点均无后继结点。>错

445、在一个链表中，表头结点和表尾结点均无前驱结点。>错

446、在一个链表中，表头结点无前驱结点，表尾结点无后继结点。>对

447、在一个链表中，表头结点无前驱结点。>对

448、在一个链表中，表头指针用来指向第一个元素结点，若表头指针的值非空，则表头和表尾结点必然不是同一个结点。>错

449、在一个链表中，表头指针用来指向第一个元素结点，若表头指针的值为空 (NULL)，则它是一个空表。>对

450、在一个链表中，表尾结点指针域的值非空 (NULL)。>对

451、在一条变量定义语句中，当同时定义多个变量时，各变量之间只能使用逗号分开。>对

452、在一条变量定义语句中，定义的变量和赋给它的初值之间只能使用等号 (=)。>对

453、在一条变量定义语句中，定义多个变量时需要使用分号隔开。>错

454、在一条变量定义语句中，可以同时定义不同基准类型的变量。>错

455、在一条变量定义语句中，可以同时定义不同类型的变量。>错

456、在一条变量定义语句中，只能定义同一类型的变量。>对

457、在一条变量定义语句中只能定义一个变量。>错

458、在一条注释语句中，除了注释内容外，不需要使用任何特定的标记符。>错

459、在一维数组的定义语句中，给数组赋初值的部分是用一对圆括号括起来的数据表。>错

460、在一维数组的定义语句中，可以给数组中每个元素赋初值。>对

461、在一维数组的定义语句中，数组名后带有一对花括号。>错

462、在一维数组的定义语句中，数组名后带有一对圆括号。>对

463、在一维数组的定义语句中，数组名后使用中括号把数组长度值括起来。>对

464、在一维数组的定义语句中，数组名前为元素类型关键字。>对

465、在一维数组的定义语句中，数组名前为元素类型关键字。>对

466、在一行中出现多条语句时，使用标记符“//”的注释语句只能为最后一条语句。>对

467、在一行中出现多条语句时，注释语句只能为最后一条语句。>错

468、在语句“typedef int* inPointer;”中，把标识符 inPointer 定义为 int* 类型。>错

469、在语句“typedef int* inPointer;”中，把标识符 inPointer 定义为 int* 的指针变量。>错

470、在语句“typedef int* inPointer;”中，把标识符 inPointer 定义为 int* 类型。>对

471、在执行“printf(“%5d”,x);”语句时，其输出的 x 的值所占用的显示宽度将大于等于 5 个字符位置。>错

472、在执行“printf(“%6d”,x);”语句时，其输出的 x 的值所占用的显示宽度将小于 6 个字符位置。>错

473、在执行“printf(“%8.2f”,x);”语句时，其输出的 x 的值所占用的显示宽度将大于等于 8 个字符位置。>错

474、在执行“printf(“%d-%d=%d”,x,y,x-y);”语句时，其输出的前两个数据值之间将采用逗号分开。>错

475、在执行“printf(“s=%s”,“abc”);”语句时，输出结果为 s=abc。>对

476、在执行“scanf(“%d%d”,&x,&y);”语句时，需要从键盘上输入 2 个整数。>错

477、在执行“typedef int DataTypeB”语句后，标识符 DataType 与保留字 int 具有完全相同的含义和作用。>对

478、在执行“printf(“%6d”,x);”语句时，其输出的 x 的值所占用的显示宽度将小于 6 个字符位置。>错

479、在执行“printf(“%8.2f”,x);”语句时，其输出的 x 的值所占用的显示宽度将大于等于 8 个字符位置。>对

480、在执行“printf(“%d-%d=%d”,x,y,x-y);”语句时，将输出 3 个数据值。>对

481、在执行“printf(“%d-%d=%d”,x,y,x-y);”语句时，其输出的前两个数据值之间将采用逗号分开。>错

482、在执行“printf(“s=%s/n”,“abc”);”语句时，输出结果为 s=abc。>对

483、在执行“scanf(“%d%d”,&x,&y);”语句时，输入的两个整数之间要用分号隔开。>错

484、在执行“scanf(“%d%d”,&x,&y);”语句时，需要从键盘上输入 2 个整数。>对

485、增量表达式 ++y 表示成赋值表达式为 y=y+1。>错

486、执行“intx=43,y=13;printf(“%d”,x%y);”语句序列后得到的输出结果为 3。>错

487、执行“printf(“%c,F-2)>错

488、执行“printf(“%c”,F-2);”语句后得到的输出结果为 H。>错

489、执行“typedef char charType;”语句后，在使用 char 定义字符变量的地方都可以使用字符类型标识符 charType 来定义字符变量。>对

490、执行“typedef int intType;”语句后，在使用 int 定义整型变量的地方都可以使用整型类型标识符 intType 来定义整型变量。>错

491、执行 calloc(n,8) 函数调用时，将动态分配得到 8*n 个字节的存储空间。>对

492、执行 malloc(sizeof(struct BB)) 函数调用时，得到的动态存储空间能够保存具有 struct BB 结构类型的一个对象。>对

493、执行“intx=43,y=13; printf(“%d”,x%y);”语句序列后得到的输出结果为 3。>错

494、执行“intx=45,y=13; printf(“%d”,x/y);”语句序列后得到的输出结果为 3。>对

495、执行“typedef char charType;”语句后，在使用 char 定义字符变量的地方都可以使用字符类型标识符 charType 来定义字符变量。>对

496、指针变量的加减，相当于指针的位移，所以只有指向数组的指针加减才是有意义的，对指向普通变量的加减没有意义。>对

497、注释内容的开始所使用的标记符为/*，则注释内容的结束所使用的标记符为*/。>对

498、注释语句的内容只会出现在源程序文件中，不会出现在编译后生成的目标代码文件中。>对

499、字符常量和字符串常量分别简称为字符和字符串，它们是分别使用单引号和双引号作为起止定界符的数据 (单词)。>对

500、字符串 aA//xxk//数据的长度为 13。>错

501、字符串“a: xxk 数据”的长度为 13。>错

502、字符串不允许为空，至少需要包含一个字符。>错

503、字符串函数 strlen 的返回值类型为整数。>对

504、字符串函数 strlen 的返回值类型为字符指针。>错

505、字符串允许为空，并且其长度为 0。>对

填空(129)--电大资源网: <http://www.dda123.cn/> (微信搜: 905080280)

- 1、C 程序中的所有预处理命令均以 () 字符开头。># (或井号)。
- 2、C 语言中的每个程序文件在编译时可能出现有 error 类型的错误，它属于 (致命性 / 警告)。
- 3、C 语言中的每条复合语句以 () 作为结束符。>《或右花括号》。
- 4、C 语言中的每条复合语句以右 () 括号作为结束符。>花 ()。
- 5、C 语言中的每条简单语句以 () 作为结束符。>: (或分号)。

C 程序中的所有预处理命令均以 () 字符开头。答案: # (或井号)。

- 6、double 类型的长度为 ()。>8
- 7、int 数据类型的长度为 ()。>4
- 8、NULL 是一个符号常量，作为空指针，它所表示的整数值为 ()。>0

9、shortint 类型的长度为 ()。>2
10、strcat () 函数用于 () 两个字符串。>连接
11、strcpy 函数用于把一个字符串 () 到另一个字符数组空间中。>拷贝 (复制)。
12、strlen () 函数用于计算一个字符串的 ()。>长度
13、trcat () 函数用于 () 两个字符串。>连接
14、把表达式 $x=x+y$ 转换成复合赋值表达式为 ()。> $x+=Y$
15、把间接访问表达式 $p=>name$ 改写成对应的直接访问表达式为 ()。> $(*p).name$
16、保留字 case 和 default 出现在 (if/switch/for/while) () 语句中。>switch
17、表达式 $25/4$ 的值为 ()。>6
18、表达式 (float) $25/4$ 的值为 6.25。>
19、表达式 (int) 1 > 14
20、程序中的每个编译单位是一个 ()。>程序文件
21、存储一个空字符串需要占用 () 个字节。>1
22、存储字符 'a' 需要占用存储器的 () 个字节空间。>1
23、存储字符串 "a" 需要至少占用存储空间中的 () 个字节。>2
24、当不需要函数返回任何值时，则应使用 () 保留字来定义函数类型。>void
25、对于 "for (i=0; i<10; i+=2) S" 语句，循环体 S 被执行的次数为 ()。>5
26、对于 "for (i=0; i<10; i++) S" 语句，循环体 S 被执行的次数为 ()。>10
27、关系表达式 $a<b$ 的相反关系式为 ()。> $a<=b$
28、很定要动态分配一个类型为 structWorker 的对象，并由 r 指针指向这个对象，则 r 的数据类型为 ()。>structWorker*
29、假定一个枚举类型的定义为 "enum RA (ab, ad, ae) 户，则 ad 的值为 ()。>2
30、假定 a 是一个一维数组，则 $a[i]$ 的指针访问方式为 ()。> $*(a+i)$ 。
31、假定 p 是一个变量，则该变量的地址表示为 ()。>&p
32、假定 p 是一个指向整数对象的指针，则用 () 表示该整数对象。>*p
33、假定 p 所指对象的值为 2>25
34、假定 p 所指对象的值为 25, p+1 所指对象的值为 46，则 'p+' 的值为 ()。>25
35、假定 P 所指对象的值为 25, p+1 所指对象的值为 46，则 *++p 的值为 ()。>46
36、假定 p 所指对象的值为 25, p+1 所指对象的值为 46，则执行表达式 (*p)++ 后，p 所指对象的值为 ()。>26
37、假定 p 所指对象的值为 25, p+1 所指对象的值为 46，则执行表达式 (P++) 后，p 所指对象的值为 ()。>46
38、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则 *++p 的值为 46。>
39、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则洛 p++ 的值为 ()。>25
40、假定 p 所指对象的值为 25，p+1 所指对象的值为 46，则执行表达式 (*p)++ 后，p 所指对象的值为 26。>
41、假定 P 所指对象的值为 34, p-1-1 所指对象的值为 4>34

42、假定 $x=5$ ，则执行 $a=(x? 10; 20)$ ；“语句后 a 的值为 ()。>10
43、假定 $x=5$ ，则表达式 $2+x++$ 的值为 7。>
44、假定 $x=5$ ，则表达式 $2+x++$ 的值为 ()。>7
45、假定 $x=5$ ，则执行 "a=(!x? 10; 20)；" 语句后 a 的值为 20。>
46、假定 $x=8$ ，则表达式 $2+x++$ 的值为 ()。>10
47、假定 x 的值为 5，则执行 "a=(!x? 10; 20)；" 语句后 a 的值为 ()。>20
48、假定 x 的值为 >8
49、假定 x 的值为 5，则执行 "a=(!x? 10; 20)；" 语句后 a 的值为 (20)。>
50、假定 $y=10$ ，则表达式 $++y*3$ 的值为 ()。>33
51、假定不允许使用逻辑非操作符，则关系表达式 $x+y>5$ 的相反表达式为 ()。> $x+y<=5$
52、假定二维数组的定义为

假定二维数组的定义为 "double a[M][N];"，则所有元素的列下标的取值范围在之间。
答案：0~N-1

53、假定二维数组的定义为 "chara[M][N];"，则数组共包含有 () 个字符元素。>M*N
54、假定二维数组的定义为 "doublea[M][N];"，则每个数组元素的行下标取值范围在 0~M-1 之间。>
55、假定二维数组的定义为 "doublea[M][N];"，则数组元素的行下标取值范围在 () 之间。>0~M-1
56、假定二维数组的定义为 "doublea[M][N];"，则所含元素的列下标的最大值为 ()。>N-1
57、假定二维数组的定义为 "doublea[M][N];"，则数组元素的行下标的最大值为 ()。>M-1
58、假定二维数组的定义为 "inta[3][5];"，则该数组所包含的元素个数为 ()。>15
59、假定二维数组的定义为 "inta[3][5];"，则该数组所占存储空间的字节数为 ()。>60
60、假定二维数组的定义为，"inta[3][5];"，则该数组所包含的元素个数为 ()。>15
61、假定要动态分配一个类型为 structWorker 的对象，并由 r 指针指向这个对象，则 r 的数据类型为 ()。>structWorker*
62、假定要动态分配一个类型为 structWorker 的对象，并由 r 指针指向这个对象，则使用的表达式为 () = malloc<sizeof (structWorker))。> *rker*r
63、假定要访问一个结构对象 x 中的数据成员 a，则表示方式为 ()。>x.a
64、假定一个结构类型的定义为 "structA (inta,b; A*c;)；"，则该类型的理论长度为 ()。>12
65、假定一个结构类型的定义为 "structB (inta[5]; char*b;)；"，则该类型的理论长度为 ()。>24
66、假定一个枚举类型的定义为 "enumRA (ab,a>1
67、假定一个枚举类型的定义为 "enumRA{ab, ac, ad, ae};"，则 ac 的值为 (1)。>
68、假定一个枚举类型的定义为则 ad 的值为 ()

假定一个枚举类型的定义为 则 ad 的值为 ()
答案：2
69、假定一个数据对象为 int* 类型，则指向该对象的指针类型为 ()。>int**
70、假定一维数组的定义为 "Char*a[8];"，则该数组所占用的存储空间的字节数为 ()。>32
71、假定一维数组的定义为 "char*a[M];"，则该数组所占存储空间的字节数为 ()。>4*M
72、假定一维数组的定义为 "char*a[8];"，则该数组所占用的存储空间的字节数为 ()。>32
73、假定一维字符指针数组的定义为 "Char*a[5];"，则该数组占用的存储空间的字节数 >20
74、假定一维字符指针数组的定义为 "Char*a [8];"，则该数组占用的存储空间的字节数为 ()。>32
75、空字符串的长度为 ()。>0
76、类型。>BB
77、逻辑表达式 $(x==0 \& \& y>5)$ 的相反表达式为 ()。> $(x!=0 \vee y<=5)$ 或： $(xy<=5)$ 。
78、每个 C 语言程序文件在编译时可能出现有致命性错误，其对应的标识符为 ()。>errorC.知'A'~'Z'的 ASCII 码为 65 - 90.当执行 "intx='C'+3;" 语句后 x 的值为 ()。答案：70
79、若 p 指向 x，则 () 与 x 的表示是等价的。>*p
80、若 $x=5, y=10$ ，则执行
81、若 $x=5, y=10$ ，则 $x!=y$ 的逻辑值为 ()。>1 (真, true)。
82、若 $x=5, y=10$ ，则错 >y 的值为 ()。>0 (假)。
83、若 $x=5, y=10$ ，则执行 "if (x<=y) c=3; elsec=8;" 语句后，c 的值为 ()。>3
84、若 $x=5, y=8$ ，则执行 "if (x>y) C=2; elseC=3;" 语句后，>3
85、若 $x=5, y=10$ ，则 $x>y$ 的值为 0 (假)。>
86、若要把一个整型指针 P 转换为字符指针，则采用的强制转换表达式为 ()。> (char*)。
87、若一个字符串的长度为 n，则存储它的字符数组的长度至少为 ()。>n+1
88、十进制数 25 表示成符合 C 语言规则的十六进制数为 ()。>0x19
89、十进制数 35 对应的八进制数为 43。>
90、十进制数 35 对应的十六进制数为 ()。>0x23
91、使用 "typedefcharBB[10][50];" 语句定义_BB_为含有 10 行 50 列的二维字符数组类型。>
92、一个函数定义由和 () 函数体两部分组成。>函数头
93、一个空字符串的长度为 ()。>0
94、用类型关键字表示十进制常数 >float
95、用类型关键字表示十进制常数 326f 的类型为 float。>
96、用于从键盘上为变量输入值的标准输入函数的函数名为 ()。>scanf
97、用于存储一个长度为 n 的字符串的字符数组的长度至少为 _n+1_。>
98、用于输出表达式值的标准输出函数的函数名是 ()。>printf

99、与结构成员访问表达式 (*fp) .score 等价的表达式为 fp->score。>
100、在#include 命令中所包含的文件，可以是头文件，也可以是_程序_文件。>
101、在#include 命令中所包含的文件，可以是头文件，也可以是 () 文件。>程序
102、在 C 语言中，一个函数定义由函数头和 () 这两个部分组成。>函数体
103、在程序中执行到 return 或返回语句时，将结束所在函数的执行过程，返回到调用该函数的位置。>
104、在函数调用中执行到 (return/break/continue/if) () 语句时，将结束所在函数的执行过程，返回到调用该函数的原来位置。>return
105、在函数外定义的变量，若没有被初始化则系统隐含对它所赋的初值为 0。>
106、在件 i, lud. 命令中所包含的文件. 可以是头文件，也可以是 () 文件。>程序
107、在所有函数定义之外定义的变量，若没有被初始化则系统隐含对它所赋的初值为 ()。>0
108、在一个 C 语言程序文件中，若要包含另外一个头文件或程序文件，则此包含命令的标识符为 ()。>#include
109、在一个程序的函数之外定义的整型变量，若没有被初始化，则系统隐含对它所赋的初值为 ()。>0
110、增量表达式 +x 表示成赋值表达式为 >y=y+1
111、整数据类型 int 的长度为 ()。>4
112、执行“intx=45,y=13; printf (“%d”,x/y);”语句序列后得到的输出结果为 ()。>3
113、执行“intx=45,y=8; printf (“%d”,x/y);”语句序列后得到的输出结果为 ()。>5
114、执行“printf (“%c”,A+2);”语句后得到的输出结果为 ()。>C
115、执行“printf (“%c”,F-2);”语句后得到的输出结果为 ()。>D
116、执行“typedefintABC[10];”语句把 ABC 定义为具有 10 个整型元素的 数组_类型。>
117、执行“typedefintABC[10];”语句把 ABC 定义为具有 10 个整型元素的 () 类型。>一维数组
118、执行“typedefintDataType;”语句后， () 与 int 保留字具有相同作用。>DataType
119、执行“typedefintabc[10];”语句把 AB>数组
120、执行 int*p=mallo (sizeof (int)) 操作得到的一个动态分配的整型对象为 ()。>*p
121、执行“intx=45,y=13; printf (“%d”, x/y);”语句序列后得到的输出结果为 (3)。>
122、执行“intx=45,y=8; printf (“%d”, x/y);”语句序列后得到的输出结果为 ()。>5
123、执行“typedefintABC[10];”语句把 ABC 定义为具有 10 个整型元素的一维 () 类型。>数组
124、执行“typedefintDataType;”语句后， () 与 int 保留字具有相同作用。>DataType
125、直接访问表达式 (*fp) .score 所对应的间接访问表达式为 (fp->score)。>

126、直接访问表达式 (*fp) .score 所对应的间接访问表达式为 ()。>fp->score
127、字符串“a: //xxk/file.txt”的长度为 ()。>15
128、字符串“a: //xxk/file.txt”的长度为_15_。>
129、作为语句标号使用的 case 和 default 只能用于 switch 或开关或分情况_语句的定义体中。>
编程题(49)--电大资源网: http://www.dda123.cn/ (微信搜: 905080280)

- 1、按照“int FF(int a[], int n)”的函数声明，编写出...
- 2、按照“int FF (inta[], intn)”的函数声明，编写出...
- 3、编写一个程序，计算 1+3+32+...+310 的值并输出...
- 4、编写一个程序，输出 50 以内 (含 50) 的、能够被 3 或...
- 5、编写一个递归函数“int FF(int a[], int n)...
- 6、编写一个函数，函数头格式为“double Mean(do...
- 7、编写一个函数，函数原型为“int fun4(char*a)...
- 8、编写一个主函数，计算 1+3+32+...+310 的值并输...
- 9、编写一个主函数，计算并输出 12+22+...+n2 值，其...
- 10、编写一个主函数，假定 a 和 b 的取值范围为: ...
- 11、编写一个主函数，利用 while 循环，求出并显示满...
- 12、编写一个主函数，利用 while 循环计算...
- 13、编写一个主函数，求出满足不等式 22+42+...+n2...
- 14、补充完善下面的一个主函数，假定函数体中定义...
- 15、补充完整下面 xxk2 函数定义，假定函数参数表中...
- 16、补充完整下面 xxk2 函数定义，假定函数参数表中...
- 17、补充完整下面 xxk2 函数定义，假定函数参数表中...
- 18、补充完整下面的函数定义，该函数要求返回满足...
- 19、补充完整下面的函数定义，该函数要求返回满足...
- 20、补充完整下面的函数定义，计算出 1+1*3+2*3+ ...
- 21、补充完整下面的函数定义，计算出整型数组 a 中 n...
- 22、补充完整下面的函数定义，利用 for 循环计算 1+2...
- 23、补充完整下面的函数定义，利用 for 循环计算 1+2...
- 24、补充完整下面的函数定义，利用 for 循环计算 1+4...
- 25、补充完整下面的函数定义，要求返回二维数组 a[...]
- 26、补充完整下面的函数定义，要求返回二维数组 a[...]
- 27、补充完整下面的函数定义，要求返回二维数组 a[...]
- 28、补充完整下面函数 xxk5 的定义，计算并返回数组...
- 29、补充完整下面函数定义中 while 循环的循环体，...
- 30、补充完整下面主函数的函数体，计算并输出表达...
- 31、补充完整下面主函数的函数体，计算并输出表达...
- 32、补充完整下面主函数的函数体，计算并输出表达...
- 33、根据函数原型“double Mean(double a[M][N])...
- 34、根据函数原型“double Mean (doublea[M][N],i...
- 35、根据函数原型“doubleMean (doublea[MIEN],in...
- 36、根据函数原型“int FF(int a[], int n)”，编...
- 37、根据函数原型“int MM(int a[], int m)”，编写...
- 38、根据函数原型“int MM(int a[], intr m)”，编...
- 39、根据函数原型“intDD (inta[], intn,intMM)”“编...
- 40、根据函数原型“intFF (inta[], int n)”，编写函数...
- 41、根据函数原型“intMM (intaE],intrn)”，编写函...
- 42、根据函数原型“void DD (int a[], int n, int MM) ...

43、完善下面程序，该程序功能是从键盘上...
44、完善下面程序中 xxk1 函数的定义，使程序能够...
45、完善下面程序中 xxk1 函数的定义，使程序能够输...
46、完善下面程序中 xxk1 函数的定义，使程序能够输...
47、完善下面程序中 xxk1 函数的定义，使程序能够输...
48、完善下面程序中 xxk1 函数的定义，使程序能够输...
49、完善下面程序中 xxk1 函数的定义，使程序能够输...
1、按照“int FF(int a[], int n)”的函数声明，编写出递归函数的定义，求出数组 a 中所有 n 个元素之积并返回。
int FF(int a[], int n)
{
if(n<=0) {printf("n 值非法\n"),exit(1);}
if(n==1) return a[0];
else return a[n-1] * FF(a,n-1);
}
2、按照“int FF (inta[], intn)”的函数声明，编写出递归函数的定义，求出数组 a 中所有 n 个元素之积并返回
35. int FF(int a[], int n)
{
if(n<=0) {printf("n 值非法\n"),exit(1);}
if(n==1) return a[0];
else return a[n-1] * FF(a,n-1);
}
3、编写一个程序，计算 1+3+32+...+310 的值并输出，假定分别用 i,p,s 作为循环变量、累乘变量和累加变量的标识符。 程序为：
#include<stdio.h>
void main(){
int i;
int p=1;
int s=1;
for(i=1;i<=10;i++){p*=3; s+=p;}
printf("%d\n",s); }
4、编写一个程序，输出 50 以内 (含 50) 的、能够被 3 或者 5 整除的所有整数。
程序为: #include<stdio.h>
void main() {
int i;
for(i=3; i<=50; i++)
if(i%3==0 || i%5==0) printf("%d ",i);
printf("\n"); }
50

5、编写一个递归函数“int FF(int a[], int n)”，求出数组 a 中所有 n 个元素之积并返回。

答案：

```
int FF(int a[], int n)
{
    if(n==1) return a[n-1];
    else return a[n-1]*FF(a,n-1);
}
```

6、编写一个函数，函数头格式为“double Mean(double a[M][N],int m,int n)”，要求返回二维数组 a[m][n]中所有元素的平均值，假定在计算过程中采用变量 v 存放平均值。

答：double Mean(double a[M][N],int m,int n)

```
{
    int i,j;
    double v=0.0;
    for(i=0; i<m; i++)
        for(j=0; j<n; j++) v+=a[i][j];
    v/=m*n;
    return v;
}
```

7、编写一个函数，函数原型为“int fun4(char*a)”，请采用 while 循环求出由字符指针 a 所指

编写一个函数，函数原型为“int fun4(char*a)”，请采用 while 循环求出由字符指针 a 所指向的字符串中包含的十进制数字的个数，并把这个值作为函数值返回。

答案：

```
int fun4(char * a)
{
    int n=0;
    while(* a) {
        if(* a>='0' && * a<='9') n++;
        a++;
    }
    return n;
}
```

8、编写一个主函数，计算 $1+3+3^2+\dots+3^{10}$ 的值并输出，假定分别用 i,p,s 作为循环变量、累乘变量和累加变量的标识符。

程序为：

```
#include<stdio.h>
void main(){
    int i;
    int p=1;
    int s=1;
```

```
for(i=1;i<=10;i++) {p*=3; s+=p;}
printf("%d\n",s); }
```

9、编写一个主函数，计算并输出 $1^2+2^2+\dots+n^2$ 值，其中 n 值由键盘输入。

答：#include<stdio.h>

```
void main()
{
    int i; //用 i 作为计数（循环）变量
    int s=0; //用 s 作为累加变量
    int n;
    printf("输入一个自然数:");
    scanf("%d",&n);
    for(i=1; i<=n; i++) s+=i*i;
    printf("s=%d\n",s);
}
```

10、编写一个主函数，假定 a 和 b 的取值范围为：

32. 编写一个主函数，假定 a 和 b 的取值范围为： $6 \leq a \leq 50, 10 \leq b \leq 30$ ，求出满足不定方程 $3a+2b=120$ 的全部整数解。如 (20,30) 就是其中的一组解。

答案：

```
void main()
{
    int a,b; //1分
    for(a=6;a<=50;a++) //2分
        for(b=10;b<=30;b++) //3分
            if(3*a+2*b==120) printf("(%d,%d)\n",a,b); //6分
}
```

11、编写一个主函数，利用 while 循环，求出并显示满足不等式的最小 n 值。

答：#include<stdio.h>

```
void main()
{
    int i=0; double s=0; //或 int i=1; double s=1;
    while(s<5) s+=(double)1/++i;
    printf("n=%d\n",i);
}
```

12、编写一个主函数，利用 while 循环计算

31. 编写一个主函数，利用 while 循环计算 $1+2+4+\dots+2^{10}$ 的值，最后输出计算结果。假定计数变量用 i 表示，每次累乘 2 的变量用 p 表示，每次累加一个数据项值的变量用 sum 表示。

答案：

```
void main()
{
    int i=1, p=1, sum=1; //1分
    while(i<=10) {p*=2; sum+=p; i++;} //4分
    printf("sum=%d\n",sum); //6分
}
```

13、编写一个主函数，求出满足不等式 $2^2+4^2+\dots+n^2 < 1000$ 的最大 n 值，假定分别用 i 和 s 作为取偶数值和累加值的变量，并限定使用 do 循环编程。

答：#include<stdio.h>

```
void main()
{
    int i=0; //用 i 作为依次取值偶数的变量
    int s=0; //用 s 作为累加变量
    do {
        i+=2;
        s+=i*i;
    }while(s<1000);
    printf("n=%d\n",i-2);
}
```

14、补充完善下面的一个主函数，假定函数体中定义的整数变量 a 和 b 的取值范围分别为：

$6 \leq a \leq 50, 10 \leq b \leq 30$ ，求出满足不定方程 $3a+2b=120$ 的全部整数解。如 (20, 30) 就是其中的一组解。

void main()

```
{
    int a,b;
    答案：
    void main ()
    {
        int a, b;
        for(a=6;a<=50;a++)//1分
        for(b=10;b<=30;b++)//3分
            if(3*a+2*b==120) printf("(%d,%d)\n", a, b); //6分
    }
```

15、补充完整下面 xxk2 函数定义，假定函数参数表中 a 的值小于等于 b，函数功能是求出并返回 a 到 b 之间（包含 a 和 b 值在内）所有偶数之和。

int xxk2(int a, int b) {

int i, sum=0;

//向下补充内容，用一个 for 循环实现

for(i=a; i<=b; i++)

```

if(i%2==0) sum+=i;
return sum;
}

```

16、补充完整下面 `xxk2` 函数定义，假定函数参数表中 `a` 的值小于等于 `b`，函数功能是求出并返回 `a` 到 `b` 之间（包含 `a` 和 `b` 值在内）所有偶数之和。

```

int xxk2(int a, int b)
{
    int i, sum=0;

    //向下补充内容,用一个 for 循环实现

    return sum;
}

```

答案:

```

for(i=a; i<=b; i++)
    if(i%2==0) sum+=i;

```

17、补充完整下面 `xxk2` 函数定义，假定函数参数表中 `a` 的值小于等于 `b`，函数功能是求出并返回 `a` 到 `b` 之间（包含 `a` 和 `b` 值在内）所有奇数之和。

```

int xxk2(int a, int b) {
int i, sum=0;
//向下补充内容, 用一个 for 循环实现
for(i=a; i<=b; i++)
if(i%2==1) sum+=i;
return sum;
}

```

18、补充完整下面的函数定义，该函数要求返回满足不等式

$1+2^2+3^2+\dots+n^2 \leq x$ 最大的值。

```

int FF (int x) {
int n=1, s=1; //使用 n 作为计数变量, 使用 s 作为累加和
while (s<=x) {n++; s+=n*n;}
//在下面一行写出合适的 while 循环语句
return n-1;
}

```

19、补充完整下面的函数定义，该函数要求返回满足不等式 $1+2^2+3^2+\dots+n^2 \leq x$ 的最大 `n` 的值。

20、补充完整下面的函数定义，计算出 $1+1*3+2*3+\dots+n*3$ 的值并返回。

```

int FF(int n) {
int i, s=1;
//在下面添上一行 for 循环语句, 循环变量为 i, 累加变量为 s
return s;
}

```

答案: `for (i=1;i<=n;i++)s+=i * 3; //4 分`
 21、补充完整下面的函数定义，计算出整型数组 `a` 中 `n` 个元素的所有偶数之和。

```

Int xxk4(int a[],int n){
Int i,s=0,
//在下面添上一行 for 循环语句, 循环变量为 i, 累加变量为 s
for(i=0;i<n;i++)if(a[i]%2==0)s+=a[i];
Return s;}

```

22、补充完整下面的函数定义，利用 `for` 循环计算 $1+21+22+22+\dots+2^n$ 的值，最后返回计算结果。

补充完整下面的函数定义，利用 `for` 循环计算 $1+21+22+\dots+2^n$ 的值，最后返回计算结果。假定计数变量用 `i` 表示，每次累乘 2 的变量用 `p` 表示，每次累加一个数据项值的变量用 `sum` 表示。

```

int xxk8(int n)
{
int i, p=1,sum=1;
//补充完整下面 for 循环语句的循环体（即一条复合语句）
for(i=1; i<=n; i++);
return sum;
}

```

答: `{p*=2;sum+=p;}`
 23、补充完整下面的函数定义，利用 `for` 循环计算 $1+21+22+\dots+2^n$ 的值，最后返回计算结果。假定计数变量用 `i` 表示，每次累乘 2 的变量用 `p` 表示，每次累加一个数据项值的变量用 `sum` 表示。

```

int xxk8(int n){
int i, p=1,sum=1;
//补充完整下面 for 循环语句的循环体（即一条复合语句）
for(i=1;i<=n;i++){p*=2;sum+=p;}
return sum;}

```

24、补充完整下面的函数定义，利用 `for` 循环计算 $1+4^1+4^2+\dots+4^m$ 的值，最后返回计算结果。

补充完整下面的函数定义，利用 `for` 循环计算 $1+4^1+4^2+\dots+4^m$ 的值，最后返回计算结果。假定计数变量用 `i` 表示，每次累乘 4 的变量用 `p` 表示，每次累加一个数据项值的变量用 `sum` 表示。

```

int xxk8 (int n)
{
int i, p=1, sum=1;
//补充完整下面 for 循环语句的循环体（即一条复合语句）
for (i=1; i<=n; i++) ()
return sum;
}
{p*=4;sum+=p;}

```

25、补充完整下面的函数定义，要求返回二维数组 `a[m][n]` 中所有元素的平均值。

补充完整下面的函数定义，要求返回二维数组 `a[m][n]` 中所有元素

的平均值。假定在计算过程中采用变量 `s` 存放累加值，采用 `i` 和 `j` 作为扫描数组的循环变量。Double Mean (double a[][N], int m, int n)

```

{
inti, j; Double s=0.0;
//在下面添上一条合适的 for 双重循环语句
return s/(m *n);
}
for(i=0;i<m;i++)
for(j=0;j<n;j++)
s+=a[i][j];

```

26、补充完整下面的函数定义，要求返回二维数组 `a[m][n]` 中所有元素的平均值。假定在计算过程中采用变量 `s` 存放累加值，采用 `i` 和 `j` 作为扫描数组的循环变量。

补充完整下面的函数定义，要求返回二维数组 `a[m][n]` 中所有元素的平均值。假定在计算过程中采用变量 `s` 存放累加值，采用 `i` 和 `j` 作为扫描数组的循环变量。Double Mean(double a[][N],int m,int n)

```

{
Int i,j; Double s=0.0;
//在下面添上一条合适的 for 双重循环语句
for(i=0;i<m;i++)
for(j=0;j<n;j++)s+=a[i][j];
Return s/(m*n);
}

```

27、补充完整下面的函数定义，要求返回二维数组 `a[m][n]` 中所有元素的平均值。假定在计算过程中采用变量 `s` 存放累加值，采用 `i` 和 `j` 作为扫描数组的循环变量。double Mean (double a[][N], int m, int n) inti, j; doubles=0.0; /在下面添上合适的 for 双重循环语句 returns/ (m*n);

28、补充完整下面函数 `xxk5` 的定义，计算并返回数组 `a[n]` 中所有元素的平均值。Double xxk5(int a[],int n){ int l,sum=0;

```

//假定分别设置 i 和 sum 为循环变量和累加变量
//在下面添上一行 for 循环语句, 循环变量为 i, 累加变量为 sum
for(i=0;i<n;i++)sum+=a[i];
Return double(sum)/n;
}

```

29、补充完整下面函数定义中 `while` 循环的循环体，该函数的功能是求出并返回由字符指针 `ptr` 所指向的字符串中包含的十进制数字字符的个数。

```

int xxk7(char* ptr) {int n=0;
//使用 n 作为统计变量
while(*ptr) {
//在下面补充合适的循环体内容
if(*ptr>='0' && *ptr<='9') n++;
ptr++;
}
return n;
}

```

30、补充完整下面主函数的函数体，计算并输出表达式 $1+2^2+3^2+\dots+10^2$ 的值。

```
#include void main() {
int i, s=0;
//假定分别设置 i 和 s 为循环变量和累加变量
//在下面添加一条 for 循环语句
for(i=1; i<=10; i++)
s+=i*i;
printf("s=%d\n",s);
}
```

31、补充完整下面主函数的函数体，计算并输出表达式 $1+2^2+3^2+\dots+20^2$ 的值。

```
#include void main() {int i, s=0;
//假定分别设置 i 和 s 为循环变量和累加变量
//在下面添加一条 for 循环语句
for(i=1; i<=20; i++)
s+=i*i;
printf("s=%d\n",s);
}
```

32、补充完整下面主函数的函数体，计算并输出表达式 $1+2^3+3^3+\dots+10^3$ 的值。

```
#include void main() {int i, s=0;
//假定分别设置 i 和 s 为循环变量和累加变量
//在下面添加一条 for 循环语句
for(i=1; i<=10; i++)
s+=i*i*i;
printf("s=%d\n",s);
}
```

33、根据函数原型“double Mean(double a[M][N],int m,int n)”，编写函数定义，要求返回二维数组 a[m][n]中所有元素的平均值。假定在计算过程中采用变量 v 存放累加值和最后的平均值。Double Mean(double a[M][N],int m,int n) {

```
int i,j;Double v=0.0;
for(i=0; i<m; i++)
for(j=0; j<n;j++) v+=a[i][j];
v/=m*n;
return v; } //注:函数体的最后两行可以合并为一条返回语句:return v/=m*n,也可以把"="去掉。
```

34、根据函数原型“double Mean (double a[M][N],int m,int n)”，编写函数定义，要求返回二维

根据函数原型"double Mean (double a[M][N],int m,int n)", 编写函数定义，要求返回二维数组 a[m][n] 中所有元素的平均值。假定在计算过程中采用变量 S 存放累加值，采用 i 和 j 作为扫描数组的循环变量。

答案: Double Mean (double a[M][N], int m, int n) {
int i+j;double s=0.0;
for (i=0;i<m;i++)
for (j=0;j<n;j++) s+=a[i][j];
return s/ (m*n) ;

35、根据函数原型“double Mean (double a[M][N],int m,int n)”，编写函数定义，要求返回二维数组 a[m][n]中所有元素的平均值。假定在计算过程中采用变量 v 存放累加值和最后的平均值。

根据函数原型"double Mean (double a[M][N],int m,int n)", 编写函数定义，要求返回二维数组 a[m][n] 中所有元素的平均值。假定在计算过程中采用变量 v 存放累加值和最后的平均值。

答案: Double Mean (double a[M][N], int m, int n)

```
{
int i,j; double v=0.0;
for (i=0; i<m; i++)
for (j=0; j<n; j++) v+=a[i][j];
v/=m*n;
return v;
}
```

36、根据函数原型“int FF(int a[], int n)”，编写函数定义，计算并返回数组 a[n]中所有元素之和。

程序为:
int FF(int a[], int n) {
int i,sum=0;
for(i=0; i<n; i++) sum+=a[i];
return sum; }

37、根据函数原型“int MM(int a[], int m)”，编写函数定义，计算并返回数组 a[m]中元素最大值和最小值之差。

```
int MM(int a[], int m){
int i,x1,x2;
x1=x2=a[0];
for(i=1; i<m;i++){
if(a[i]>x1) x1=a[i];
if(a[i]<x2) x2=a[i];}
return x1-x2;}
```

38、根据函数原型“int MM(int a[], intr m)”，编写函数定义，计算并返回数组 a[m]中所有元素

根据函数原型"int MM(int a[], intr m)", 编写函数定义，计算并返回数组 a[m]中所有元素的最大值的平方值，假定用变量 max 保存最大值。

答案:
int MM(int a[],int m){
int i,max=a[0];
for(i-1;i<m;i++)
if(a[i]>max)max=a[i]
return max * max;
}

39、根据函数原型“intDD (inta[], intn,intMM)”，编写函数定义，利用双重 for 循环统计出数组 a[n]

根据函数原型"intDD (inta[], intn,intMM)"编写函数定义，利用双重 for 循环统计出数组 a[n]中任何两个不同元素的值等于 MM 值的配对个数，然后返回统计结果.如假定 a[对+a[j]等于 MM，同时 i≠j，则为一个配对。

intDD (inta[], intn,intMM)

```
{
inti,j, C=0; //用 i 和 i 作为循环变量，用 C 作为统计变量
//在下面补充一个双重 for 循环语句
Return C;
}
```

答案: for (i=0; i<n; i++) //1 分
for (j=i+1; j<n; j++) //2 分

if (a[i]+a[j]==MM) C++; //6 分

40、根据函数原型“intFF (inta[], int n)”，编写函数定义，计算并返回数组 a[n]中对中所有元素之和。

答案: int FF(int a[],int n){
int i, sum=0;
for(i=0;i<n;i++)sum+=a[i];
return sum;
}

41、根据函数原型“intMM (intaE[],intrn)”，编写函数定义，计算并返回数组 aEm) 中元素最大值和最小值之差。

根据函数原型"intMM (intaE[],intrn)", 编写函数定义，计算并返回数组 aEm) 中元素最大值和最小值之差。

答案:
int MM(int a[],int m){
int i,x1,x2; x1=x2=a[0];
for(i=1; i<m; i++){
if(a[i]>x1) x1=a[i];
if(a[i]<x2) x2=a[i];
}
return x1-x2;
}

42、根据函数原型“void DD (int a[], int n, int MM)”，编写函数定义，利用双重 for 循环统计出数组 a[n]中任何两个不同元素的值等于 MM 值的配对个数，然后返回统计结果。如假定 a[i]+a[j]

等于 MM，同时 itj，则为一个配对。

答案:
int DD(int a[],int n,int MM){
int i,j,c=0;
for (i=0; i<n; i++)
for (j=i+1;j<n;j++)
if (a[i]+a[j]==MM) c++;
return c;
}

43、完善下面程序，该程序功能是从键盘上

31. 完善下面程序,该程序功能是从键盘上输入 10 个整数到一维数组 a[10]中,下标从大到小的次序输出数组 a[10]中的每个元素值,元素值之间用一个空格分开。

```
#include<stdio.h>

void main()
{
    int a[10],i; //用 i 作为循环变量
    printf("从键盘输入 10 个整数: \n");
    //向下补充程序内容,用两个 for 循环实现
    printf("\n");
}
```

答案:

```
//向下补充程序内容,用两个 for 循环实现
for(i=0; i<10; i++) scanf("%d",&a[i]); //3 分
for(i=9; i>=0; i--) printf("%d ",a[i]); //6 分
```

注:第 1 行中的 &a[i]也可替换为 a+i。

44. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的最大值。#include<stdio.h> int xxk1(int a[], int n); void main ()

45. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的总和。

```
#include int xxk1(int a[], int n);
void main() {int x;
int a[6]={ 15,20,36,48,24,44};
x=xxk1(a,6);
printf("x=%d\n", x);
} int xxk1(int a[], int n) {int i, sum=a[0];
//向下补充内容,用一个 for 循环实现
for(i=1; i<n; i++) sum+=a[i];
return sum;
}
```

46. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的最大值。

```
#include int xxk1(int a[], int n);
void main() {
int x;
int a[6]={ 15,20,36,48,24,44};
x=xxk1(a,6);
printf("x=%d\n", x);
} int xxk1(int a[], int n) {
int i, max=a[0];
//向下补充内容,用一个 for 循环实现
for(i=1; i<n; i++) max=a[i];
return max;
}
```

47. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的最小值。

```
#include int xxk1(int a[], int n);
void main() {int x;
int a[6]={ 15,20,36,48,24,44};
x=xxk1(a,6);
printf("x=%d\n", x);
} int xxk1(int a[], int n) {int i, min=a[0];
//向下补充内容,用一个 for 循环实现
for(i=1; i<n; i++) if(a[i]<min) min=a[i];
return min;
}
```

48. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的最大值。

```
#include<stdio.h>
int xxk1(int a[], int n);
void main()
{
int x;
int a[6]={ 15,20,36,48,24,44};
x=xxk1(a,6);
printf("x=%d\n", x);
}
int xxk1(int a[], int n)
{
int i, x=a[0], max=a[0];
//向下补充内容,用一个 for 循环实现
```

答案:

```
for(i=1; i<n; i++)
if(a[i]>max) max=a[i];
```

49. 完善下面程序中 xxk1 函数的定义,使程序能够输出数组 a 中所有元素的最大值。

```
#include<stdio.h>
int xxk1 (int a [], int n);
Void main (){
int x;
int a [6] = { 15,20,36,48,24,44};
x=xxk1(a,6);
printf("x=%d \n", x);}
int xxk1(int a[], int n){
```

```
int i, x=a [0], max=a [0];
//向下补充内容, 用一个 for 循环实现
for(i=1;i<n;i++)
if(a[i]>max) max=a[i];
return max;
```

程序填空题(230)--电大资源网: <http://www.dda123.cn/> (微信搜: 905080280)

1. #include<stdio.h> Void main () { intx=15...
2. # include<stdio.h>void mmin () {int m=10,v0...
3. # include<stdio.h>void main () {int x=5,v=...
4. #include void main () { int a[8]={3,5,7,9,...
5. #include int Count(int a[], int n, int x...
6. #include int LA(int *a, int n, int x) {
7. #include int WF(int x, int y) {
8. #include struct Worker {
9. #include struct Worker {
10. #include struct Worker {char name[15]; i...
11. #include void main () {int i,s=0;for(i=16; ...
12. #include#define N 4 void main () {int a[N];...
13. #include#includevoid main () {char ch;...
14. #include<stdio.h> int LA(int * a, i...
15. #include<stdio.h>
16. #include<stdio.h> int LA(int *a, int n) {...
17. #include<stdio.h> int WF(int x, int y)...
18. #include<stdio.h> void main () { int a[3 ...
19. #include<stdio.h> void main () { int a[8]=...
20. #include<stdio.h> void main () { int i,j ...
21. #include<stdio.h> void main () { int x=20;...
22. #include<stdio.h># define N 6
23. #include<stdio.h># define M 6
24. #include<stdio.h># define M 6
25. #include<stdio.h># define N 6
26. #include<stdio.h>#define M 6
27. #include<stdio.h>#define N 6
28. #include<stdio.h>#define N 6 void main () {...
29. #include<stdio.h>#define N6 void main () {i...
30. #include<stdio.h>#defineN6 voidmain () {int...
31. #include<stdio.h>#define N6
32. #include<stdio.h>#include<string.h>...
33. #include<stdio.h>#include<string.h>...
34. #include<stdio.h>#include<string.h> str...
35. #include<stdio.h>#include<string.h> voi...
36. #include<stdio.h>#include<string.h> voi...
37. #include<stdio.h> const int M=20;
38. #include<stdio.h> const int M=20;
39. #include<stdio.h> const int M=20;
40. #include<stdio.h> int a[10]={4,5,6,15,20...
41. #include<stdio.h> int a[6]= {4,5,6,15,20...
42. #include<stdio.h> int a[6]={4,5,6,15,20,...
43. #include<stdio.h> int Count(int a[], int...

44. [#include<stdio.h> int LA\(int * a, int n\)...](#)
45. [#include<stdio.h> int LA\(int *a, int n\) ...](#)
46. [#include<stdio.h> int LA\(int *a, int n, ...](#)
47. [#include<stdio.h> int LB\(int * a, int n\)...](#)
48. [#include<stdio.h> int LB\(int *a, int n\) ...](#)
49. [#include<stdio.h> int WF\(int x, in: y\) {...](#)
50. [#include<stdio.h> int WF\(int x, int y\) {...](#)
51. [#include<stdio.h> int WF\(int x, int y\) {...](#)
52. [#include<stdio.h> int WF\(int x, int y\) {...](#)
53. [#include<stdio.h> main \(\) {inta=1,b=3,c=5;...](#)
54. [#include<stdio.h> struct Worker \(char na...](#)
55. [#include<stdio.h> struct Worker \(char na...](#)
56. [#include<stdio.h> struct Worker {](#)
57. [#include<stdio.h> struct Worker {](#)
58. [#include<stdio.h> struct Worker { char n...](#)
59. [#include<stdio.h> struct Worker {char na...](#)
60. [#include<stdio.h> struct Workerichar nam...](#)
61. [#include<stdio.h> struer Worker {char na...](#)
62. [#include<stdio.h> void LE\(int * a, int *...](#)
63. [#include<stdio.h> void LE\(int* a, int* b\)...](#)
64. [#include<stdio.h> void main \(\) { int n=4,y...](#)
65. [#include<stdio.h> void main \(\) {char s\[\]=...](#)
66. [#include<stdio.h> void main \(\) {int i, f1=...](#)
67. [#include<stdio.h> void main \(\)](#)
68. [#include<stdio.h> void main \(\)](#)
69. [#include<stdio.h> void main \(\) {...](#)
70. [#include<stdio.h> void main \(\) {...](#)
71. [#include<stdio.h> void main \(\) {...](#)
72. [#include<stdio.h> void main \(\) {...](#)
73. [#include<stdio.h> void main \(\) {...](#)
74. [#include<stdio.h> void main \(\) {...](#)
75. [#include<stdio.h> void main \(\) {...](#)
76. [#include<stdio.h> void main \(\) { char a \[\]=...](#)
77. [#include<stdio.h> void main \(\) { char a \[\]=...](#)
78. [#include<stdio.h> void main \(\) { char a \[\]=...](#)
79. [#include<stdio.h> void main \(\) { int a,b;...](#)
80. [#include<stdio.h> void main \(\) { int a\[10\]=...](#)
81. [#include<stdio.h> void main \(\) { int a\[3\]\[3\]...](#)
82. [#include<stdio.h> void main \(\) { int a\[6\]=...](#)
83. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
84. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
85. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
86. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
87. [#include<stdio.h> void main \(\) { int a\[8\]=1...](#)
88. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
89. [#include<stdio.h> void main \(\) { int a\[9\]=...](#)
90. [#include<stdio.h> void main \(\) { int a\[9\]=...](#)
91. [#include<stdio.h> void main \(\) { int a, b;...](#)
92. [#include<stdio.h> void main \(\) { int f1,f2,...](#)
93. [#include<stdio.h> void main \(\) { int i, s1=...](#)
94. [#include<stdio.h> void main \(\) { int i, s=0...](#)
95. [#include<stdio.h> void main \(\) { int i, s=0...](#)
96. [#include<stdio.h> void main \(\) { int i,j,k=...](#)
97. [#include<stdio.h> void main \(\) { int i,j,k=...](#)
98. [#include<stdio.h> void main \(\) { int i,s=0;...](#)
99. [#include<stdio.h> void main \(\) { int i,s=0;...](#)
100. [#include<stdio.h> void main \(\) { int i,s=0;...](#)
101. [#include<stdio.h> void main \(\) { int i;...](#)
102. [#include<stdio.h> void main \(\) { int i, s=0;...](#)
103. [#include<stdio.h> void main \(\) { int n=4, v...](#)
104. [#include<stdio.h> void main \(\) { int n=6, v...](#)
105. [#include<stdio.h> void main \(\) { int x=20...](#)
106. [#include<stdio.h> void main \(\) { int x=20...](#)
107. [#include<stdio.h> void main \(\) { int x=5;...](#)
108. [#include<stdio.h> void main \(\) { struct Wor...](#)
109. [#include<stdio.h> void main \(\) {int i, s=0;...](#)
110. [#include<stdio.h> void main \(\) {int i,j,k=0...](#)
111. [#include<stdio.h> void main \(\) {int i,s=0;...](#)
112. [#include<stdio.h> void main \(\) {int x=5, v=8...](#)
113. [#include<stdio.h> void main \(\) { int a\[3\]\[3\]...](#)
114. [#include<stdio.h> void main \(\) {int a \[8\] = ...](#)
115. [#include<stdio.h> Void main \(\) {Int a\[8\]={2...](#)
116. [#include<stdio.h> void main \(\) {int i=0,s=0;...](#)
117. [#include<stdio.h> Void main \(\) {Int x=5,y=15...](#)
118. [#include<stdio.h> Void main \(\) {inti,f=1;...](#)
119. [#include<stdio.h> Void main \(\) {intx=15, v=8,z...](#)
120. [#include<stdio.h> void SB\(char ch\) {...](#)
121. [#include<stdio.h> void SB\(char ch\) { swi...](#)
122. [#include<stdio.h> void SB\(char ch\)](#)
123. [#include<stdio.h> voidmain \(\) {chardata\[5\];...](#)
124. [#include<stdio.h> voidmain \(\) {intx=2,v=-1,z...](#)
125. [#include<stdio.h> Voidmain \(\) 。 Intx=5,v=15...](#)
126. [#include<stdio.h> void main \(\) 。 int a\[3\]...](#)
127. [#include<stdio.h> void main \(\) 。 int i=0,...](#)
128. [#include<stdio.h>void main \(\) {int a\[3\]\[3\]={...](#)
129. [#include<stdio, h> void main \(\) { int a\[3\]\[3...](#)
130. [#include<stdio, h>void main \(\) {int a\[8\]={10...](#)
131. [#include<stdio, h> # define N 8 void m...](#)
132. [#include<stdio, h> void main \(\) { int a\[3...](#)
133. [#include<stdio, h> void main \(\) { int x=5...](#)
134. [#include<stdio, h> void main \(\) {int a\[3\]...](#)
135. [#include<stdio, h> void main \(\) {int a\[3\]...](#)
136. [#include<stdio, h> void main \(\) {int a\[8\]...](#)
137. [#include<stdio, h> Void main \(\) { Int a8...](#)
138. [#include<stdio, h> void main \(\) {int i=0...](#)
139. [#include<stdio, h> void main \(\) {int i=1 ...](#)
140. [#include<stdio, h> Void main \(\) { inti, f=...](#)
141. [#include<stdio, h> void main \(\) { int x=5 ...](#)
142. [#include<stdio, h> void main \(\) { int x=5,v...](#)
143. [#include<stdio, h> void main \(\) { int a\[3\]\[3...](#)
144. [#include #define M6](#)
145. [#include<stdio.h> #define N 4 void main...](#)
146. [#include #define NS](#)
147. [#include<stdio.h> #defineN5 void main \(\) {...](#)
148. [#include<stdio.h> #void main \(\) { inti, fl=1...](#)
149. [#include<stdio.h> struct Worker{char na...](#)
150. [#include<stdio.h> void main \(\) { inta~25, b=...](#)
151. [#include<Stdio.h> void main \(\) { inta\[8\]=...](#)
152. [#include<stdio.h> void main \(\) { inta \(8\) ...](#)
153. [#include<stdio.h> void main \(\) { intn=0, v=0;...](#)
154. [#include<stdio.h> Void main \(\) {int n=10, y=0...](#)
155. [#include<stdio.h> void main \(\) {inta \(8\) = {...](#)
156. [#include # define N 8](#)
157. [#include<stdio.h> #define N 5 void main \(\) ...](#)
158. [#include<stdio.h> void main \(\) { int x=12,...](#)
159. [#include<stdio.h> void main \(\) { int a\[8\]=...](#)
160. [#include<stdio.h> void main \(\) { int a\[8\] =...](#)
161. [#include<stdio.h> void main \(\) { int i,f1=1...](#)
162. [#include<stdio.h> void main \(\) { int x=5 ,v...](#)
163. [#include<stdio.h> void main \(\) { int x=5,v=...](#)
164. [#include<stdio.h> void main \(\) { int x=5,y=...](#)
165. [#include<stdio.h> void main \(\) { inti, s1=0...](#)
166. [#include<stdio.h> void main \(\) { intx=1,v; ...](#)
167. [#include<stdio.h> void main \(\) { int a=5, b=8, c...](#)
168. [#include<stdio.h> void main \(\) { int a\[4\]\[3\]...](#)
169. [#include<stdio.h> Void main \(\) { int x=5,y=8...](#)
170. [#include<stdio.h> void main \(\) { int x=5, v=15...](#)
171. [#include<stdio.h> void main \(\) { int x=5, v=8, z...](#)
172. [#include<stdio.h> void main \(\) { inta\[6\] = {...](#)
173. [#include<stdio.h> void main \(\) { inti, s=0;...](#)
174. [#include<stdio.h> Void main \(\) {int a\[8\]={2...](#)
175. [#include<stdio.h> void main \(\) {int i=0, s=0; ...](#)
176. [#include<stdio.h> void main \(\) {inti=0, s=0;...](#)
177. [#include void SB\(char ch\){](#)
178. [#include<stdio.h> voidmain \(\) { int n=2, y=0; wh...](#)
179. [#include void main \(\) 。](#)
180. [#includevoid main \(\) {char ch;](#)
181. [#includevoid main \(\) {int a=25,b=18;](#)
182. [#includevoid main \(\) {int a=5,b=8,c;](#)
183. [#includevoid main \(\) {int a=5,b=8,c=6,d=15,...](#)
184. [#includevoid main \(\) {int a\[8\]={10,8,12,15,...](#)
185. [#includevoid main \(\) {int a\[8\]={10,8,16,15,...](#)
186. [#includevoid main \(\) {int a\[8\]={10,8,6,15,9...](#)
187. [#includevoid main \(\) {int a\[8\]={3,8,6,5,9,2...](#)
188. [#includevoid main \(\) {int i, s1=0, s2=0;...](#)
189. [#includevoid main \(\) {int i, s=0;](#)
190. [#includevoid main \(\) {int i,f,f1=1,f2=2;...](#)
191. [#includevoid main \(\) {int i,s=0;](#)
192. [#includevoid main \(\) {int i,s=0;](#)
193. [#includevoid main \(\) {int i,s=0;](#)

194、#include void main () {int i,s=0;
 195、#include void main () {int i,x,y,s=0;
 196、#include void main () {int i=0, s=0;
 197、#include void main () {int x=5,y=0;
 198、#include void main () {int x=5,y=8,z=12;...
 199、#include void main () {int x=5,y=8,z=14,w;...
 200、#include void main () {int x=5,y=8,z=4;...
 201、#include void main () {int x=5,y=8,z=6,w=0;...
 202、#include void main () {int x=5,y=8;
 203、#include void main () {int x=9,v=0;
 204、include<stdio. h>void main(
 205、include<stdio.h> void main () { int x=5...
 206、include<stdio.h> void main () {int f1=1,f2;...
 207、include<stdio.h>define N6
 208、include<stdio.h>int WF(intx,inty){
 209、include<stdio.h>void main () { int a,bv...
 210、include<stdio.h>void main () { int i;...
 211、include<stdio.h>void main () { int i,j,k=0;...
 212、include<stdio.h>voidLE(inta,intb){
 213、include<stdio.h>voidmain () {int i,s=0;...
 214、include<stdio.h>[int WF(intx,inty)]{...
 215、includeconstintM=20
 216、includeinta[10]={4,5,6,15,20,13,12,7,8,9...
 217、includevoidfun(charss[])
 218、includevoidmain ()
 219、includevoidmain ()
 220、includevoidmain () {
 221、includevoidmain () {
 222、includevoidmain () {
 223、includevoidmain () {
 224、includevoidmain () {
 225、int xxk3(int n){int i,s=1,
 226、intmain () {inti,sum=0;
 227、void main () { int a[3][3]={1(3,5,7),(9,11;...
 228、当使用 xkZC 调用此函数时，返回值为() ...
 229、假定使用 xxk3(5)调用下面函数，则返回值为 () 。 i...
 230、假定使用 xxk3 (4) 调用下面函数，则返回值为 () ...
 1、**#include Void main () { intx=15, y=8,z=4;**
include<stdio. h>
Void main(){
intx=15, y=8,z=4;
if(x>y)x--y;
if (x>z)x-z ;
printf("%d/n" , x);
}
答案： 4
 2、**# include<stdio. h>void mnin () {int m=10,y0;**
while(a>5) {
y+nn-
printi("y=%d/n",y)
 行下面程序的输出结果为 (**y=34**)

3、**# includevoid main () {int x=5,y=8;**
include<stdio. h>
void main() {
int x=5,y=8;
if(x>=y) printf("%d %d\n",x,y);
else printf("%d %d\n",y,x);
}
 执行下面程序的输出结果为 (**8 5**) 。
 4、**#include void main () { int a[8]={3,5,7,9,2,3,4,8};**
#include
void main() {
int a[8]={3,5,7,9,2,3,4,8};
int s=0,*p;
for(p=a;p<a+8;) s+=*p++;
printf("s=%d\n",s);
}
答案： s=41
 5、**#include int Count(int a[], int n, int x)**
#include
int Count(int a[], int n, int x)
{
int i,c=0;
for(i=0;i<n;i++) if(a>x) c++;
return c;
}
void main() {
int a[8]={20,15,32,47,24,36,28,70};
int b=Count(a,8,30);
printf("b=%d\n",b);
}
答案： B=4
 6、**#include int LA(int *a, int n, int x) {**
#include
int LA(int *a, int n, int x) {
int i,s=0;
for(i=0;i<n;i++)
if(a<x) s+=a;
return s;
}
void main() {
int a[8]={5,10,15,8,12,3,9,20};
int b=LA(a,5,10);
int c=LA(a+2,6,10);
printf("%d %d\n",b,c);
}
答案： 13 20
 7、**#include int WF(int x, int y) {**
#include
int WF(int x, int y) {
x=x+y;
y=x+y;
}

return x+y;
}
void main() {
int x=5, y=7;
int z=WF(x,y);
printf("z=%d\n",z);
}
答案： z=31
 8、**#include struct Worker {**
#include
struct Worker {Char name[15]; //姓名
int age; //年龄
float pay; //工资
};
void main() {
struct Worker x={wanghua,52,4300};
struct Worker y, *p;
y=x; p=&x;
printf("%s %d %6.2f\n,x.name,y.age,p->pay);
}
答案： wanghua 52 4300
 9、**#include struct Worker {**
#include
struct Worker {Char name[15]; //姓名
int age; //年龄
float pay; //工资
};
void main() {
struct Worker x;Char *t=liouting;
int d=38; float f=3493;
strcpy(x.name,t);
x.age=d; x.pay=f;
printf("%s %d %6.0f\n,x.name,x.age,x.pay);
}
答案： liouting 39 2493
 10、**#include struct Worker {char name[15]; int age; float pay;};**
#include
struct Worker {char name[15]; int age; float pay;};
int Less(struct Worker r1, struct Worker r2) {
if(r1.age<r2.age) return 1; else return 0;
}
void main() {
struct Worker a[4]={{abc,25,2420},{def,58,4638},
{ghi,49,4260},{jkl,36,3750}};
struct Worker x=a[0];
int i;
for(i=1; i<4; i++)
if(Less(x,a) x=a;
printf("%s %d %6.0f\n,x.name,x.age,x.pay);
}
答案： def 58 4638

```

11、#include void main () {int i,s=0;for(i=16; i>0; i-=2)
#include void main() {
    int i,s=0;
    for(i=16; i>0; i-=2)
        if(i%3==0 || i%4==0) s+=i;
    printf("s=%d\n",s);
}

```

执行下面程序的输出结果为 (s=46)。

```

12、#include#define N 4 void main () {int a[N];
    int i;
    a[0]=1;
    for(i=1;

```

i
执行下面程序的输出结果为 (A.15)。

```

13、#include#includevoid main () {char ch;
    int a,b,c;
    double d;
    scanf("%c",&ch);
    scanf("%d %d %d",&a,&b,&c);
    switch(ch) {case 'r': d=a+b+c;Break;
    case 's': d=a*b*c;Break;
    case 't': d=(a+b+c)/3. 0;Break;
    default: {printf("Error!\n");

```

```

    exit(1);
    } } printf("d=%Lf\n",d);
}

```

执行下面程序时，假定从键盘上输入的字符为 t，输入的 3 个整数为 2、5 和 8，则程序输出结果为 (A.d=5. 0)。

```

14、#include<stdio. h> int LA(int * a, int n){
#include<stdio. h>
    int LA(int * a, int n){
        int i, s=0;
        for(i=0;i<n; i++)s+=a[i]
        return s;
    }

```

```

void main(){
    int a[5] = {1, 2, 3, 4, 5};
    int b=LA(a,5)+LA(a+1, 3)
    printf("b=%An", b);
}

```

答案: b=24

```

15、#include<stdio. h>
#include<stdio. h>

```

```

void main(){
    int x=20;
    int i=2;
    while(i<x){
        if(x%i==0){printf"%d", i);x/=i;}
        i++;
    }
}

```

```

}
答案: 2 5
16、#include<stdio. h> int LA(int * a, int n) {
#include<stdio. h>
int LA(int * a, int n) {
    int i,s=0;
    for(i=0; i<n;i++) s+=a[i];
    return s; }
void main() {
    int a[5]={1 ,2,3,4,5};
    int b=LA(a,5)+LA(a+2, 3);
    printf("b=%d\n",b);}
运行结果: b=27

```

```

17、#include<stdio. h> int WF(int x , int y) {
#include<stdio. h>
int WF(int x , int y) {
    x=x+y;
    y+=x;
    return x+y; }
void main() {
    int x=3 , y=8;
    printf("%d\n" ,WF(x,y));}
运行结果: 30

```

```

18、#include<stdio. h> void main () { int a[ 3 J [ 31 = { { 3, 8, 6} ,
{ 5, 9, 2} , { 8, 5, 12} } ;

```

```

#include<stdio. h>
void main() {
    int a[3][3]={{3,8,6},{5,9,2},{8,5,12}};
    int i,j,s=0;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            if(a[i][j]%3==0) s+=a[i][j];
    printf("s= %d\n",s);
}

```

答: s=30

```

19、#include<stdio. h> void main () { int
a[8]={76,63,54,95,40,75,90,66};
#include<stdio. h>
void main() {
    int a[8]={76,63,54,95,40,75,90,66};
    int i, s=0;
    for(i=0;i<8;i++)
        if(a[i]>=70 && a[i]<=90) s+=a[i];
    printf("s=%d\n",s) ; }
运行结果: s=241

```

```

20、#include<stdio. h> void main () { int i,j ,k==0;
#include<stdio. h>
void main() {
    int i,j ,k==0;
    for ((i=0;i<5;i++)
        for(j=i;j<5;j++) k++;;
    printf("%d\n",k) ; }
运行结果: 15

```

```

21、#include<stdio. h> void main () { int x=20;
#include<stdio. h>

```

```

void main() {
    int x=20;
    int i=2;
    while(i<x) {
        if(x%i==0) {printf("%d",i); x/=i;}
        i++; } }
运行结果: 25

```

```

22、#include<stdio. h># define N 6

```

```

#include<stdio. h>
#define N 6
void main() {
    int a[N];
    int i;
    a[0]=1; a[1]=2;
    for(i=2; i<N; i++) a[i]=a[i-2]+a[i-1];
    printf("%d\n",a[N-1]);
}

```

执行下面程序的输出结果为 (13) .

```

23、#include<stdio.h> # define M 6

```

```

27. # include<stdio. h>
#define M 6
void main()
{
    int i, x1=0, x2=0;
    int a[M] = {10, 15, 22, 24, 12, 30};
    for(i=0; i<M/2; i++) {x1+=a[i]; x2+=a[M-i-1];}
    printf("%d%d\n", x1, x2);
}

```

输出结果:

答案: 47 66

24、#include<stdio.h> #define M 6

```
27. #include<stdio.h>
```

```
#define M 6
```

```
void main()
```

```
{
```

```
int i,x1=0,x2=0;
```

```
int a[M]={2,5,8,4,6,9};
```

```
for(i=0;i<M/2;i++){x1+=a[i];x2+=a[M-i-1];}
```

```
printf("%d%d\n",x1,x2);
```

```
}
```

输出结果: 15 19

答案: 15 19

25、#include<stdio.h> #define N 6

```
#include<stdio.h>
```

```
#define N 6
```

```
void main()
```

```
{
```

```
int i,a[N]={2,5,8,10,15,21};
```

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

```
if(a[i]%5==0) printf("%d",a[i]);
```

```
printf("\n");
```

```
}
```

答案: 5 10 15

26、#include<stdio.h> #define M 6

```
#include<stdio.h>
```

```
#define M 6
```

```
void main() {
```

```
int i,x;
```

```
int a[M]={10,15,22,37,46,58};
```

```
for(i=0; i<M/2; i++)
```

```
{x=a[i]; a[i]=a[M-1-i]; a[M-1-i]=x;}
```

```
for(i=0; i<6; i++) printf("%d",a[i]);
```

```
printf("\n"); }
```

运行结果: 58 46 37 22 15 10

27、#include<stdio.h> #define N 6

```
#include<stdio.h>
```

```
#define N 6
```

```
void main() {
```

```
int i, a[N]={2, 5, 8, 10, 15, 21};
```

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

```
if(a[i]%5) printf("%d ", a[i]);
```

```
printf("\n"); }
```

运行结果: 2 8 21

28、#include<stdio.h> #define N 6 void main () {int a[N];

```
int i:A[0]=1;a[1]=2;
```

```
for(i=2;i<N;i++)a[i]=a[i-2]+a[i-1];
```

```
printf("%d\n",a[N-1]);
```

```
}
```

执行下面程序的输出结果为(13)。

29、#include<stdio.h> #define N6 void main () {int a[N];

```
int i:A[0]=1;a[1]=2;
```

```
for(i=2;
```

```
i<N;
```

```
i++)a[i]=a[i-2]+a[i-1]+1;
```

```
printf(" %d\n ",a[N- 1]);
```

```
}
```

执行下面程序的输出结果为(20)。

30、#include<stdio.h> #define N6 voidmain ()

```
{inti,a[N]={2,5,8,10,15,21};
```

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

```
if (a[i]%5) printf ("%d",a[i]) ;
```

```
printf ("\n") ;
```

```
}
```

请写出上面程序的运行结果: (B.2821)。

31、#include<stdio.h> #define N6

执行下面程序的输出结果为 ()。

```
#include<stdio.h>
```

```
#define N6
```

```
void main ()。
```

```
int a[N];
```

```
int i: A[0]=1; a[1]=2;
```

```
for (i=2; i<N; i++) a[i]=a[i-2]+a[i-1]+1;
```

```
printf (" %d\n " ,a[N- 1])。
```

```
)
```

答: 20

32、#include<stdio.h> #include<string.h>

```
#include<stdio.h>
```

```
#include<string.h>
```

```
void fun(char ss[]);
```

```
void main() {
```

```
char s[15]="0123456789";
```

```
fun(s);
```

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

```
}
```

```
void fun(char ss[]) {
```

```
int i, n=strlen(ss) ;
```

```
for(i=0; i<n/2; i++) {
```

```
char c=ss[i];
```

```
ss[i]=ss[n-1-i];
```

```
ss[n-1-i]=c;
```

```
}
```

答案: 9876543210

33、#include<stdio.h> #include<string.h>

```
#include<stdio.h>
```

```
#include<string.h>
```

```
struct Worker {
```

```
char name[15]; //姓名
```

```
int age; //年龄
```

```
float pay; //工资
```

```
};
```

```
void main() {
```

```
struct Worker x;
```

```
char *t="liouting";
```

```
int d=38; float f=3493;
```

```
strcpy(x.name,t);
```

```
x.age=d; x.pay=f;
```

```
printf("%s %d %6.0f\n",x.name,x.age,x.pay);
```

```
}
```

答案: liouting 39 2493

34、#include<stdio.h> #include<string.h> struct Worker { char

```
name[15]; int age; float pay;};
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
struct Worker { char name[15]; int age; float pay;};
```

```
void main() {
```

```
struct Worker x;
```

```
char *t="liouting";
```

```
int d=38; float f=400;
```

```
strcpy(x.name,t);
```

```
x.age=d; x.pay=f;
```

```
x.age++; x.pay*=2;
```

```
printf("%s %d %6.2f\n",x.name,x.age,x.pay); }
```

运行结果: liouting 39 800.00

35、#include<stdio.h> #include<string.h> void main() {

```
#include<stdio.h>
```

```
#include<string.h>
```

```
void main() {
```

```
char s[15]="567891234";
```

```
int i, n=strlen(s) ;
```

```
for(i=0; i<n/2; i++) {
```

```
char c=s[i];
```

```
s[i]=s[n-1-i];
```

```
s[n-1-i]=c; }
```

```
printf("%s\n",s); }
```

运行结果: 432198765

36、#include<stdio.h> #include<string.h> void main () { int i;

```
#include<stdio.h>
```

```
#include<string.h>
```

```
void main() {
```

```
int i;
```

```

unsigned int len;
char* a[5]={"student", "worker", "cadre", "soldier",
"zzeasan123"};
len=strlen(a[0]);
for(i=1; i<5; i++)
    if(strlen(a[i])>len) len=strlen(a[i]);
printf("%d\n", len);

```

运行结果: 10

37、**#include<stdio.h> const int M=20;**

```

25. #include<stdio.h>

const int M=20;

void main() {
    int i=2, k=0;
    while(i<M) {
        if (M%i==0) k++;
        i++;
    }
    printf("k=%d\n",k);
}

```

输出结果:↵

答案: k=4 ↵

38、**#include<stdio.h> const int M=20;**

```

#include<stdio.h>
const int M=20;
void main() {
    int i=2;
    while(1) {
        if(i>M/2) break;
        if(M%i==0) printf("%d ",i);
        i++; }
    printf("\n"); }

```

运行结果: 2 4 5 10

39、**#include<stdio.h> const int M=20;**

```

#include<stdio.h>
const int M=20;
void main()
{
    int i,c2,c3,c5;
    c2=c3=c5=0;
    for(i=1; i<=M; i++) {
        if(i%2==0) c2++;
        if(i%3==0) c3++;

```

```

if(i%5==0) c5++;
    }
    printf("%d %d %d\n",c2,c3,c5);
}

```

答案: ①10 6 4

40、**#include<stdio.h> int a[10]={4,5,6,15,20,13,12,7,8,9};**

```

#include<stdio.h>
int a[10]={4,5,6,15,20,13,12,7,8,9};
void main()
{
    int i,s[4]=0;
    for(i=0; i<10; i++) {
        switch(a[i]%4) {
            case 0: s[0]+=a[i];break;
            case 1: s[1]+=a[i];break;
            case 2: s[2]+=a[i];break;
            default: s[3]+=a[i];break;
        }
    }
    for(i=0; i<4; i++) printf("%d ",s[i]);
}

```

答案: ①44 27 6 22

41、**#include<stdio.h> int a[6]= {4,5,6,15,20,12};**

```

26. #include<stdio.h>

int a[6]= {4,5,6,15,20,12};

void main()
{
    int i,s1,s2;
    s1=s2=0;
    for(i=0; i<6; i++)
        if(a[i]%3) s1+=a[i]; else s2+=a[i];
    printf ("%d%d\n", s1, s2);
}

```

输出结果:↵

答案: 29 33 ↵

42、**#include<stdio.h> int a[6]={4,5,6,15,20,12};**

```

#include<stdio.h>
int a[6]={4,5,6,15,20,12};
void main() {
    int i,s1,s2;
    s1=s2=0;
    for(i=0; i<6; i++) {

```

```

switch(a[i]%2) {
    case 0: s2+=a[i];break;
    case 1: s1+=a[i];break; } }
printf("%d %d\n",s1,s2); }

```

运行结果: 20 42

43、**#include<stdio.h> int Count(int a[], int n, int x)**

```

#include<stdio.h>
int Count(int a[], int n, int x)
{
    int i,c=0;
    for(i=0;i<n;i++)
        if(a[i]>x) c++;
    return c;
}
void main() {
    int a[8]={20,15,32,47,24,36,28,70};
    int b=Count(a,8,30);
    printf("b=%d\n",b);
}

```

答案: B=4

44、**#include<stdio.h> int LA(int * a, int n) {**

```

5. #include<stdio.h>

int LA(int * a, int n) {
    int i,s=0;
    for(i=0;i<n;i++) s+=a[i];
    return s;
}

void main() {
    int a[5]={1,2,3,4,5};
    int b=LA(a,5)+LA(a+2,3);
    printf("b=%d\n",b);
}

```

答案: b=27 ↵

45、**#include<stdio.h> int LA(int *a, int n) {**

```

#include<stdio.h>
int LA(int *a, int n) {
    int i,s=0;
    for(i=0;i<n;i++) s+=a[i];
    return s; }
void main() {
    int a[5]={1,2,3,4,5};
    int b=LA(a,5)+LA(a+1,3);
    printf("b=%d\n",b); }

```

运行结果: b=24

```
46、#include<stdio.h> int LA(int *a, int n, int x) {
#include<stdio.h>
int LA(int *a, int n, int x) {
int i,s=0;
for(i=0;i<n;i++)
if(a[i]<x) s+=a[i];
return s;
}
void main() {
int a[8]={5,10,15,8,12,3,9,20};
int b=LA(a,5,10);
int c=LA(a+2,6,10);
printf("%d %d\n",b,c);
}

```

答案: 13 20

47、#include<stdio.h> int LB(int *a, int n)

```
28. #include<stdio.h>
int LB(int *a, int n) {
int *p,s=1;
for(p=a;p<a+n;p++) s*=*p;
return s;
}
void main() {
int a[]={1,2,3,4,5};
int b=LB(a,5);
printf("b=%d\n",b);
}

```

输出结果:

答案: b=120

48、#include<stdio.h> int LB(int *a, int n) {

```
#include<stdio.h>
int LB(int *a, int n) {
int i,s=1;
for(i=0;i<n;i++) s*=*a++;
return s;
}
void main() {
int a[]={1,2,3,4,2,4,5,2};
int b=LB(a,4)+LB(a+3,3);
printf("b=%d\n",b);
}

```

运行结果: b=56

49、#include<stdio.h> int WF(int x, int y) {

```
27. #include<stdio.h>
int WF(int x, int y) {
x=x+y;
y+=x;
return x+y;
}
void main() {
int x=3, y=5;
printf("%d\n",WF(x,y));
}

```

答案: 21

50、#include<stdio.h> int WF(int x, int y) {

```
4. #include<stdio.h>
int WF(int x, int y) {
x=x+y;
y+=x;
return x+y;
}
void main() {
int x=3, y=8;
printf("%d\n",WF(x,y));
}

```

答案: 30

51、#include<stdio.h> int WF(int x, int y) {

```
#include<stdio.h>
int WF(int x, int y) {
x=x+y;
y=x+y;
return x+y;
}
void main() {
int x=5, y=7;
int z=WF(x,y);
printf("z=%d\n",z);
}

```

答案: z=31

52、#include<stdio.h> int WF(int x, int y) { x=x+y;

```
#include<stdio.h>
int WF(int x, int y) {
x=x+y;

```

```
y+=x;
return x+y;
}
void main() {
int x=3, y=5;
printf("%d\n",WF(x,y));
}

```

运行结果: 21

53、#include<stdio.h> main () {inta=1,b=3,c=5;

```
if(c==a+b)
printf("yes\n");
else
printf("no\n");

```

以下程序的输出为 (B.no)

54、#include<stdio.h> struct Worker (char name[15]; int age; double pay);

```
28. #include<stdio.h>
struct Worker {char name[15]; int age; double pay;};
void main() {
struct Worker a[4]={{ "abc",20,2400 },{ "defa",50,3600 },
{ "ghin",40,1800 },{ "jkl",36,2000 }};
int i; double s=0;
for(i=0;i<4;i++)s+=a[i].pay;
printf("s=%5.0lf\n",s);
}

```

输出结果:

答案: s=9800

55、#include<stdio.h> struct Worker (char name[15]; int age; float pay):

```
28. #include<stdio.h>
struct Worker {char name[15]; int age; float pay;};
void main() {
struct Worker x={"wanghua",48,2500};
struct Worker y, *p;
y=x; p=&x;
printf("%d %7.2f\n", y.age+p->age, p->pay+60);
}

```

输出结果:

答案: 96 2560.00

56、#include<stdio.h> struct Worker {

```
#include<stdio.h>
struct Worker {
char name[15]; int age; float pay; };

```

```
void main() {
    struct Worker x={"wanghua",52,2350};
    struct Worker y, *p;
    y=x; p=&x;
    printf("%d %7.2f\n", y.age+p->age, p->pay+20);
}
运行结果: 104 2370.00
```

57、**#include<stdio.h> struct Worker {**

```
#include<stdio.h>
struct Worker {
    char name[15]; //姓名
    int age; //年龄
    float pay; //工资
};
void main() {
    struct Worker x={"wanghua",52,4300};
    struct Worker y, *p;
    y=x; p=&x;
    printf("%s %d %6.2f\n",x.name,y.age,p->pay);
}
```

答案: wanghua 52 4300

58、**#include<stdio.h> struct Worker { char name[15]; int age; double pay;};**

```
5. #include<stdio.h>
struct Worker { char name[15]; int age; double pay;};
void main() {
    struct Worker a[4]={{ "abc",20,2420},{ "defa",50,3638},
        {"ghin",40,2560},{ "jkl",36,2375}};
    int i, s=0;
    for(i=0; i<4; i++) s+=a[i].age;
    printf("s=%5.2Lf\n", (double)s/4);
}
```

答案: s=36.50

59、**#include<stdio.h> struct Worker {char name[15]; int age; float pay;};**

```
#include<stdio.h>
struct Worker {char name[15]; int age; float pay;};
int Less(struct Worker r1, struct Worker r2) {
    if(r1.age<r2.age) return 1; else return 0;
}
void main() {
    struct Worker a[4]={{ "abc",25,2420},{ "def",58,4638},
        {"ghi",49,4260},{ "jkl",36,3750}};
    struct Worker x=a[0];
    int i;
    for(i=1; i<4; i++)
        if(Less(x,a[i])) x=a[i];
}
```

```
printf("%s %d %6.0f\n",x.name,x.age,x.pay);
}
```

答案: def 58 4638

60、**#include<stdio.h> struct Worker{char name[15]; int age; float pay;**

```
28. #include<stdio.h>
struct Worker{char name[15];int age;float pay;};
void main(){
    struct Worker x={"wanghua",48,2500};
    struct Worker y, *p;
    y=x;p=&x;
    printf("%d\n",y.age+p->age);
}
```

输出结果: 4

答案: 96

61、**#include<stdio.h> struct Worker {char name[15]; int age; float pay;}**

```
28. #include<stdio.h>
#include<string.h>
struct Worker {char name[15]; int age; float pay;};
void main() {
    struct Worker x;
    char *t="liouting";
    int d=38; float f=400;
    strcpy(x.name,t);
    x.age=d;x.pay=f;
    x.age--;x.pay/=2;
    printf("%s%d%6.2f\n",x.name,x.age,x.pay);
}
```

输出结果: 4

答案: liouting 37 200.00

62、**#include<stdio.h> void LE(int * a, int * b) {**

```
#include<stdio.h>
void LE(int * a, int * b) {
    int x= * a;
    * a= * b; * b=x;
}
void main() {
    int x=15, y=26;
    LE(&x,&y);
    printf("%d %d\n",x,y);
}
```

答案: 26 15

63、**#include<stdio.h> void LE(int* a, int* b) {**

```
#include<stdio.h>
void LE(int* a, int* b) {
    int x=*a;
    *a=*b; *b=x; }
void main() {
    int x=15, y=26;
    printf("%d %d\n", x, y);
    LE(&x, &y);
    printf("%d %d\n", x, y); }
```

运行结果: 15 26 26 15

64、**#include<stdio.h> void main () { int n=4,y=0;**

```
24. #include<stdio.h>
void main(){
    int n=4,y=0;
    while(n-->y+=3;
    printf("y=%d\n",y);
}
```

输出结果:

答案: y=12

65、**#include<stdio.h> void main () {char s[]="123456789"1**

```

27. #include<stdio.h>
#include<string.h>
void main() {
    char s[]="123456789";
    int i, n=strlen(s);
    for(i=0; i<n/2; i++) {
        char c=s[i]; s[i]=s[n-1-i]; s[n-1-i]=c;
    }
    printf("%s\n",s);
}

```

输出结果:

答案: 987654321

```

66. #include<stdio.h> void main () {int i, f1=1, f2;
for (i=2;i<5;i++) {
f2=3*f1;
f1=f2;
}
printf("f2=%d\n",f2);
}

```

执行下面程序的输出结果为(f2=27)。

```

67. #include<stdio.h> void main ()

```

```

#include<stdio.h>
void main()
{
    int x=5;
    switch(2*x-3) {
        case 4: printf("%d ",x);
        case 7: printf("%d ",2*x+1);
        case 10: printf("%d ",3*x-1); break;
        default: printf("%s","default\n");
    }
}

```

答案: 11 14

```

68. #include<stdio.h> void main ()

```

```

#include<stdio.h>
void main()
{
    int i,s;
    for(i=1,s=0;i<10;i++) {
        if(i%3==0) continue;
        s+=i;
    }
    printf("%d\n",s);
}

```

答案: ①27

```

69. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    int i, s1=0, s2=0;
    for(i=0;i<10;i++)
        if(i%2) s1+=i; else s2+=i;
    printf("%d %d\n",s1,s2);
}

```

答案: 25 20

```

70. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    int n=6,y=1;
    while(n) y*=n--;
    printf("y=%d\n",y);
}

```

答案: ①y=720

```

71. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    int i,s=0;
    for(i=1;i<=5;i++) s+=i*i;
    printf("s=%d\n",s);
}

```

答案: s=55

```

72. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    int a[8]={12,39,26,41,55,63,72,40};
    int i, i1=0, i2=0;
    for(i=0;i<8;i++)
        if(a[i]%2==1) i1++; else i2++;
    printf("%d %d\n",i1,i2);
}

```

答案: ①4 4

```

73. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    char a[]="abcdbfbgacd";
    int i1=0, i2=0, i=0;
    while(a[i]) {
        if(a[i]=='a') i1++;
        if(a[i]=='b') i2++;
        i++;
    }
    printf("%d %d %d\n",i1,i2,i);
}

```

答案: ①2 3 11

```

74. #include<stdio.h> void main () {
#include<stdio.h>

```

```

void main() {
int a[3][4]={{1,2,7,8},{5,6,10,6},{9,12,3,4}};
int m=a[0][0];
int i,j;
for(i=0; i<3; i++)
for(j=0; j<4; j++)
if(a[i][j]>m) m=a[i][j];
printf("%d\n",m);
}

```

答案: ①12

```

75. #include<stdio.h> void main () {
#include<stdio.h>
void main() {
    int a[8]={3,5,7,9,2,3,4,8};
    int s=0,*p;
    for(p=a;p<a+8;) s+=*p++;
    printf("s=%d\n",s);
}

```

答案: s=41

```

76. #include<stdio.h> void main () { char a []=\ "acdbfkdmd\ "

```

```

26. #include<stdio.h>
void main() {
    char a[]="acdbfkdmd";
    int i1=0, i2=0, i=0;
    while(a[i]) {
        if(a[i]<'d') i1++; else i2++;
        i++;
    }
    printf("%d%d\n",i1,i2);
}

```

输出结果:

答案: 3 5

```

77. #include<stdio.h> void main () { char a[]
=\ "acdbfdkmdaceg\ "

```

```

26. #include<stdio.h>
void main(){
    char a[]="acdbfdkmdaceg";
    int i=0,k=0;
    while(a[i])
        if(a[i++]=='d')k++;
    printf("%d\n",k);
}

```

输出结果:↵
 答案: 3 ↵

```

78. #include<stdio.h> void main () { char a[]="acdbfkgamd";

```

```

3. #include<stdio.h>
void main() {
    char a[]="acdbfkgamd";
    int i1=0, i2=0, i=0;
    while(a[i]) {
        if(a[i]<'e') i1++; else i2++;
        i++;
    }
    printf("%d %d\n",i1,i2);
}

```

答案: 7 4 ↵

```

79. #include<stdio.h> void main () { int a,b;

```

```

#include<stdio.h>
void main()
{
    int a,b;
    for(a=2,b=3; a<15;) {
        printf("%d %d ",a,b);
        a=a+b;
        b=a+b;
    }
}

```

答案: 2 3 5 8 13 21 ↵

```

80. #include<stdio.h> void main () { int
a[10]={12,39,26,41,55,63,72,40,83,95};
#include<stdio.h>
void main() {
    int a[10]={12,39,26,41,55,63,72,40,83,95};
    int i, i1=0, i2=0;
    for(i=0;i<10;i++)
        if(a[i]%2==1) i1++; else i2++;
    printf("%d %d\n",i1,i2); }

```

运行结果: 6 4

```

81. #include<stdio.h> void main () { int
a[3][3]={{3,5,7},{9,11,13},{6,8,20}};
#include<stdio.h>
void main() {
    int a[3][3]={{3,5,7},{9,11,13},{6,8,20}};
    int i,*p=&a[0][0];
    for(i=0;i<9;i++) {
        if(*p>10) printf("%d ",*p);
        p++; }
    printf("\n"); }

```

运行结果: 11 13 20

```

82. #include<stdio.h> void main () { int a[6]= (36,25,48,24,55,40);

```

```

25. #include<stdio.h>
void main(){
    int a[6]={36,25,48,24,55,40};
    int i=0,x=24;
    while(a[i]! =x && i<6)i++;
    if(i<6)printf("%d\n",i);
}

```

输出结果: ↵
 答案: 3 ↵

```

83. #include<stdio.h> void main () { int a[8]=
(12,39,26,41,55,63,72,40);

```

```

26. #include<stdio.h>
void main() {
    int a[8]={12,39,26,41,55,63,72,40};
    int i, c1=0, c2=0;
    for(i=0; i<8; i++)
        if(a[i]%2==1)c1++; else c2++;
    printf("%d,%d\n",c1,c2);
}

```

输出结果:
 答案: 4,4 ↵

```

84. #include<stdio.h> void main () { int a[8]=
[4,8,12,16,20,24,28,32];

```

27. #include<stdio. h>

```
void main() {
    int a[8]={4,8,12,16,20,24,28,32};
    int *p=a,s=0;
    do {s+=*p;p+=3;}while(p<a+8);
    printf("s=%d\n",s);
}
```

输出结果:

答案: s=48

85. #include<stdio.h> void main () { int a[8]={4,8,12,16,20, 24, 28,32};

4. #include<stdio. h>

```
void main() {
    int a[8]={4,8,12,16,20,24,28,32};
    int *p=a;
    do {
        printf("%d ", *p);
        p+=3;
    } while(p<a+8);
    printf("\n");
}
```

答案: 4 16 28

86. #include<stdio.h> void main () { int a[8]={76,63,54,95,40,75,90,66};

3. #include<stdio. h>

```
void main() {
    int a[8]={76,63,54,95,40,75,90,66};
    int i, s=0;
    for(i=0; i<8; i++)
        if(a[i]>=70 && a[i]<=90) s+=a[i];
    printf("s=%d\n",s);
}
```

答案: s=241

87. #include<stdio.h> void main () { int a[8]={17.6,4.5,8.3,9,12};

26. #include<stdio. h>

```
void main() {
    int a[8]={7.6,4.5,8.3,9,12};
    int i, s=0;
    for(i=0; i<8; i++)
        if(a[i]>=6 && a[i]<=9) s+=a[i];
    printf("s=%d\n",s);
}
```

答案: s=30

88. #include<stdio.h> void main () { int a[8]={70,63,54,95,40,75,90,66}; #include<stdio.h>

```
void main() {
    int a[8]={70,63,54,95,40,75,90,66};
    int i, s=0;
    for(i=0; i<8; i++)
        if(a[i]>=70 && a[i]<=90) s+=a[i];
    printf("s=%d\n",s);
}
```

运行结果: s=235

89. #include<stdio.h> void main () { int a[9]={36,25,48,24,55,40,18,66,20};

25. #include<stdio. h>

```
void main(){
    int a[9]={36,25,48,24,55,40,18,66,20};
    int i=0, k=-1, x=40;
    while(a[i]!=x && i<9)i++;
    if(a[i]==x) k=i;
    printf("k=%d\n",k);
}
```

输出结果:

答案: k=5

90. #include<stdio.h> void main () { int a[9]={36,25,48,24,55,40,18,66,20}; #include<stdio.h>

```
void main() {
    int a[9]={36,25,48,24,55,40,18,66,20};
    int i, b1, b2;
    b1=b2=a[0];
    for(i=1; i<9; i++) {
```

```
if(a[i]>b1) b1=a[i];
if(a[i]<b2) b2=a[i]; }
printf("%d %d\n",b1,b2); }
```

运行结果: 66 18

91. #include<stdio.h> void main () { int a, b; #include<stdio.h>

```
void main() {
    int a, b;
    for(a=2, b=3; b<20; ) {
        printf("%d %d ", a, b);
        a=a+b;
        b=a+b; }
    printf("%d %d\n", a, b); }
```

运行结果: 2 3 5 8 13 21

92. #include<stdio.h> void main () { int f1,f2,i; #include<stdio.h>

```
void main() {
    int f1,f2,i;
    f1=1;
    printf("%d ",f1);
    for(i=2;i<=5;i++) {
        f2=3*f1+1;
        printf("%d ",f2);
        f1=f2; }
    printf("\n"); }
```

运行结果: 1 4 13 40 121

93. #include<stdio.h> void main () { int i, s1=0, s2=0; #include<stdio.h>

```
void main() {
    int i, s1=0, s2=0;
    for(i=0;i<10;i++)
        if(i%2) s1+=i;
        else s2+=i;
    printf("%d %d\n",s1,s2); }
```

运行结果: 25 20

94. #include<stdio.h> void main () { int i, s=0;

24. #include<stdio. h>

```
void main() {
    int i, s=0;
    for(i=1;s<15;i++)
        if(i%2==0) s+=i;
    printf("s=%d\n",s);
}
```

输出结果:

答案: s=20

```
95、#include<stdio.h> void main () { int i, s=0;
#include<stdio.h>
void main() {
int i, s=0;
for(i=1;i<=10;i++) {
if(s>30) break;
if(i%2==0) s+=i;
printf("s=%d\n",s); }
运行结果: s=42
```

```
96、#include<stdio.h> void main () { int i,j,k=0
```

```
24. #include<stdio.h>
```

```
void main() {
int i,j,k=0;
for(i=0; i<=10; i++)
for(j=i; j<=10; j++) k++;
printf("%d\n",k);
}
```

答案: 10

```
97、#include<stdio.h> void main () { int i,j,k=0;
#include<stdio.h>
void main() {
int i,j,k=0;
for(i=0; i<5; i++)
for(j=i; j<5; j++) k++;
printf("%d\n",k); }
运行结果: 15
```

```
98、#include<stdio.h> void main () { int i,s=0;
```

```
#include<stdio.h>
void main() {
int i,s=0;
for(i=1;i<=6;i++) s+=i*i;
printf("s=%d\n",s);
}
```

答案: s=55

```
99、#include<stdio.h> void main () { int i,s=0;
```

```
25. #include<stdio.h>
```

```
void main()
{
int i,s=0;
for(i=1;i<=15;i++)
if(i%4==0)s+=i;
printf("s=%d\n",s);
}
```

输出结果:

答案: =24

```
100、#include<stdio.h> void main () { int i,s=0;
```

```
24. #include<stdio.h>
```

```
void main(){
int i,s=0;
for(i=1,s<=10;i++)
if(i%2==0)s+=i;
printf("s=%d\n",s);
}
```

输出结果:

答案: s=12

```
101、#include<stdio.h> void main () { int i;
```

```
#include<stdio.h>
#include<string.h>
void main() {
int i;
unsigned int len=0;
char * a[5]={"student","worker","cadre","soldier","n123"};
for(i=0; i<5; i++)
if(strlen(a[i])>len) len=strlen(a[i]);
printf("len=%d\n",len);
}
```

答案: len=7

```
102、#include<stdio.h> void main () { int i, s=0;
```

```
#include<stdio.h>
void main() {
```

```
int i, s=0;
for(i=1;i<=6;i++) s+=i*i;
printf("s=%d\n", s); }
```

运行结果: s=55

```
103、#include<stdio.h> void main () { int n=4, y=0
```

```
24. #include<stdio.h>
```

```
void main() {
int n=4, y=0;
while(n-->0) y+=3;
printf("y=%d\n",y);
}
```

输出结果:

答案: y=12

```
104、#include<stdio.h> void main () { int n=6, y=1;
```

```
1. #include<stdio.h>
```

```
void main() {
int n=6, y=1;
while(n-->0) y+=3;
printf("y=%d\n",y);
}
```

| }

答案: y=19

```
105、#include<stdio.h> void main () { int x=20
```

```
25. #include<stdio.h>
```

```
void main() {
int x=20;
int i=2;
while(i<=x) {
if(x%i==0) {printf("%d ", i); x/=i;}
else i++;
}
}
```

答案: 2 2 5

106. #include<stdio.h> void main () { int x=20

2. #include<stdio.h>

```
void main() {
    int x=20;
    int i=2;
    while(i<x) {
        if(x%i==0) {printf("%d ",i); x/=i;}
        i++;
    }
}
```

答案: 25

107. #include<stdio.h> void main () { int x=5;

#include<stdio.h>

```
void main() {
    int x=5;
    switch(2*x-1) {
        case 4: printf("%d ",x); break;
        case 7: printf("%d ",2*x); break;
        case 10: printf("%d ",3*x); break;
        default: printf("%s ", "default"); }
    printf("%s\n", "switch end."); }
运行结果: default switch end
```

108. #include<stdio.h> void main () { struct Worker
a[4]={{\ "abe",20,2400}, {\ "defa",50,3600},

28. #include<stdio.h>

```
struct Worker { char name[15]; int age; double pay; };
void main() {
    struct Worker a[4]={{ "abc",20,2400}, {"defa",50,3600},
        {"ghin",40,1800}, {"jkl",36,2200}};
    int i; double s=0;
    for(i=0; i<4; i++) s+=a[i].pay;
    printf("s=%f\n",s);
}
```

输出结果:

答案: s=1000.00

109. #include<stdio.h> void main () {int i, s=0;

25. #include<stdio.h>

```
void main()
{
    int i, s=0;
    for(i=1;i<8;i++)
        if(i%2==0 || i%3==0) s+=i;
    printf("s=%d\n",s);
}
```

输出结果:

答案: s=15

110. #include<stdio.h> void main () {int i,j,k=0

1. #include<stdio.h>

```
void main() {
    int i,j,k=0;
    for({i=0; i<5; i++})
        for(j=i; j<5; j++) k++;
    printf(" %d\n",k);
}
```

答案: 15

111. #include<stdio.h> void main () {int i,s=0;

24. #include<stdio.h>

```
void main() {
    int i, s=0;
    for(i=1;i<10;i+=2) s+=i;
    printf("s=%d\n",s);
}
```

输出结果:

答案: s=25

112. #include<stdio.h> void main () {int x=5, y=8,z=4;
if(x>y)x=y;

if(x>z)x=z;

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

执行下面程序的输出结果为(4)。

113. #include<stdio.h> void main () { int a[3][3]={{13, 18, 16}

#include<stdio.h>

void main () {

int a[3][3]={{13, 18, 16}, {15, 19, 12}, {8, 15, 22}};

int i, j, x=a[0][0];

for (i=0; i<3; i++) for (j=0; j<3; j++) if (a[i][j]<x) x=a[i][j];

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

运行结果是: x=8

114. #include<stdio.h> void main () {int a [8] =

{10,8,12,15,9,27,8,16};

int i, x=a [0];

for (i=1;i<8;i++)

if(a[i]>x)x=a[i];

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

执行下面程序的输出结果为(x=27)。

115. #include<stdio.h> void main () {Int a[8]={2,5,6,8,4,9,3,6};

Int i, x=a[0]+a[1];

for (i=1;i<7;i++) {

int y=a[i]+a[i+1];

if(y>x)x=y;

}

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

执行下面程序的输出结果为(14)。

116. #include<stdio.h> void main () {int i=0,s=0;

while(s<30){i+=2;s+=i*i;}

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

}

执行下面程序的输出结果为(i=6)。

117. #include<stdio.h> void main () {Int x=5,y=15, z=12;

if(x>y&& x>z)printf(" %d\n ",x);

elseif(y>x&& y>z)printf(" %d\n ",y);

elseprintf(" %d\n ",z);

}

执行下面程序的输出结果为(15)。

118. #include<stdio.h> void main () {inti,f=1;

for (i=1;i<4;i++) f=3*f+1;

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

}

执行下面程序的输出结果为(f=40)。

119. #include<stdio.h> void main () {intx=15, y=8,z=4;

if(x>y)x--y;

if(x>z)x-z;

printf("%d\n", x); }

执行下面程序的输出结果为(4)。

120. #include<stdio.h> void SB(char ch) {

26. #include<stdio. h>

```
void SB(char ch) {
    switch(ch) {
        case 'A': case 'a':
            printf("1"); break;
        case 'B': case 'b':
            printf("2"); break;
        default:
            printf("3"); break;
    }
}
```

```
void main() {
    char a1='a',a2='B',a3='f';
    SB(a1);SB(a2);SB(a3);
    printf("\n");
}
```

输出结果:

答案: 123

121、#include<stdio.h> void SB(char ch) { switch(ch) {

```
#include<stdio.h>
void SB(char ch) {
    switch(ch) {
        case 'A': case 'a':
            printf("WW "); break;
        case 'B': case 'b':
            printf("GG "); break;
        default:
            printf("BB "); break; } }
```

```
void main() {
    char a1='a',a2='B',a3='f';
    SB(a1);SB(a2);SB(a3);
    printf("\n"); }
```

运行结果: WW GG BB

122、#include<stdio.h> void SB(char ch)

26. #include<stdio. h>

```
void SB(char ch) {
    switch(ch) {
        case 'A': printf("1"); break;
        case 'B': printf("2"); break;
        default: printf("3"); break;
    }
}
```

```
void main() {
    char a1='a',a2='B',a3='A';
    SB(a1);SB(a2);SB(a3);
}
```

输出结果: 4

答案: 321

123、#include<stdio.h> voidmain () {chardata[5];
inti;
for (i=0;i<5;i++) Data[i]='A'+i;
for (i=0;i<5;i=i+2) Data[i]=data[i]+32;
for (i=0;i<5;i++)
printf ("%d", data[i]);
}

上面程序的运行结果是: (D.aBcDe)。

124、#include<stdio.h> voidmain () {intx=2,y=-1,z=5;

```
if(x<y)
if(y<0)
z=0;
else
z=z+1;
printf("%d\n",z);
```

以下程序的输出为 (A.5)

125、#include<stdio.h> Voidmain () 。 Intx=5,y=15, z=12;

执行下面程序的输出结果为 ()。

```
#include<stdio.h>
Voidmain ()。
Intx=5,y=15, z=12;
if (x>y&&x>z) printf ( " %d\n " ,x)。
elseif (y>x&&y>z) printf ( " %d\n " ,y)。
elseprintf ( " %d\n " ,z)。
)
```

答: 15

126、#include<stdio.h> void main () 。 int a[3][3]

执行下面程序的输出结果为 ()。

```
#include<stdio.h>
void main ()。
int a[3][3]= ((3,8,6) , (5,9,2) , (8,5,12));
int i,j,s=0;
for (i=0; i<3; i++)。
for (j=0; j<3; j++)。
if (a[i][j]>6) s+=a[i][j];
printf ( " s=%d\n " , s)。
)
```

答: s=37

127、#include<stdio.h> void main () 。 int i=0,s=0;
执行下面程序的输出结果为 ()。

```
#include<stdio.h>
void main ()。
int i=0,s=0;
while (s<30) (i+=2; s+=i*i;)
printf ( " i=%d\n " ,i)。
)
```

答: i=6

128、#include<stdio.h>void main () {int a[3][3]=
{3,8,6},{5,9,2},{8,5,12}};

int i,j,s=0;

```
for(i=0;i<3;i++)
for(j=0;j<3;j++)
if(a[i][j]>6)s+=a[i][j];
printf( " s=%d\n " , s);
```

执行下面程序的输出结果为(s=37)。

129、#include void main () { int a[3][3]={3,8,6} , {5,9,2} , {8,5,12}};

```
# include<stdio. h>
```

```
void main() {
    int a[3][3]={{3,8,6},{5,9,2},{8,5,12}};
    int i,j,s=0;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            if(a[i][j]%3==0) s+=a[i][j];
    printf("s= %d\n",s);
}
```

执行下面程序的输出结果为()。

答案: s=30

130、#includevoid main () {int a[8]={10,8,16,15,9,21,7,16};

```
#include<stdio.h>

void main() {
    int a[8]={10,8,16,15,9,21,7,16};
    int i,x=a[0];
    for(i=1;i<8;i++)
        if(a[i]<x) x=a[i];
    printf("x=%d\n",x);
}
```

答案: x=7

```
131、#include # define N 8 void main () {
#include<stdio.h>
#define N 8
void main(){
    int a[N]={3, 5, 8, 10, 15, 20, 32, 66};
    int l, x;
    for(i=0; i<N/2; i++) {
        x=a[i]; a[i]=a[N-1-i]; a[N-1-i]=x;
    }
    printf(" %d\n" ,a[1]+a[3]);
}
```

执行下面程序的输出结果为()。

答案: 47

```
132、#include void main () { int a[3][3]={{2 , 3 , 4}, {5 , 6 , 8}, {1 , 5 , 2}};
#include<stdio.h>
void main () {
    int a[3][3]={{2 , 3 , 4}, {5 , 6 , 8}, {1 , 5 , 2}};
    int * p=(int *)a;
    int s=0;
    while( p<(int *)a+9) s+= *p ++;
    printf(" %d\n" ,s);
}
```

执行下面程序的输出结果为()。

答案: 36

```
133、#include void main () { int x=5 ,y=0;
#include<stdio.h>
void main () {
    int x=5 ,y=0;
    switch(2 * x-3) {
        case 4: y +=x; break;
        case 7: y+=2 * x+1; break;
        case 10: y+=3 * x; break;
    }
    printf("y= %d\n" ,y);
}
```

执行下面程序的输出结果为 ()。

答案: y=11

```
134、#include void main () {int a[3][3]={{3 , 8 ,6},{ 5 ,9 ,2} , {8 , 5 ,12} } ;
#include<stdio.h>
void main () {
    int a[3][3]={{ 3 , 8 ,6} , { 5 ,9 ,2} , { 8 , 5 ,12} } ;
    int i ,j ,s=0;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            if( a[i][j]>6)s+=a[i][j];
    printf("s= %d\n" ,s);
}
```

执行下面程序的输出结果为()。

答案: s=37

```
135、#include void main () {int a[3][3]={{3 , 8 , 6} , {5 , 9 , 2} , {8 , 5 ,12}};
#include<stdio.h>
void main () {
    int a[3][3]={{3 , 8 , 6} , {5 , 9 , 2} , {8 , 5 ,12}};
    int i ,j ,s=0;
    for(i=0;i<3;i ++)
        for(j =0 ;j<3 :j++)
            if(a[i][j]%3= =0) s+=a[i][j] ;
    printf( "s= %d\n" ,s);
}
```

执行下面程序的输出结果为()。

答案: s=30

```
136、#include void main () {int a[8]={10 , 8 , 16 , 15 , 9 , 21 , 7 ,16};
#include<stdio.h>
void main () {
    int a[8]={10 , 8 , 16 , 15 , 9 , 21 , 7 ,16};
    int i ,x=a[0];
    for(i=1;i<8;i ++)
        if(a[i]<x) x=a[i];
    printf("x= %d\n" ,x);
}
```

执行下面程序的输出结果为()。

答案: x=7

```
137、#include Void main () { Int a[8]={2,5,6,8, 4, 9, 3,6};
#include<stdio.h>
Void main() {
    Int a[8]={2,5,6,8, 4, 9, 3,6};
    Int i, x=a[0]+a[1];
    for (i=1: i<7; i++) {
        int y=a[i]+a[i+1];
        if(y>x)x=y;
        printf("%d\n",x);
    }
}
```

答案: 14

```
138、#include void main () {int i=0 , s=0;
#include<stdio.h>
void main () {
    int i=0 , s=0;
    while(s<30) {i+=2; s+=i * i; }
    printf("s= %d\n" ,s);
}
```

执行下面程序的输出结果为()。

答案: s=56

```
139、#include void main () {int i=1 ,s=0,
#include<stdio.h>
void main() {
    int i=1 ,s=0,
    do {s+=i*i; i++;} while(s<30);
    printf( "s= %d\n" ,s);
}
```

执行下面程序的输出结果为()。

答案: s=30

```
140、#include Void main () { inti, f=1;
#include<stdio.h>
Void main(){
    inti, f=1;
    for (i=1: i<4: i++) f=3*f+1;
    printf("f=%d\n", f);
}
```

答案: f=40

```
141、#include void main () { int x=5 ,y=0;
#include<stdio.h>
void main() {
    int x=5 ,y=0;
    switch(2*x-3) {
        case 4: y+=x; break;
        case 7: y+=2*x+1; break;
        case 10: y+=3*x; break;
    }
    printf("y= %d\n" ,y);
}
```

答案: y=11

```
142、#include void main () { int x=5,y=0;
```

```
#include<stdio. h>
void main() {
    int x=5,y=0;
    switch(2 * x+3) {
        case 4: y+=x; break;
        case 7: y+=2 * x+1; break;
        case 10: y+=3 * x; break;
        default: y+=x-1;
    }
    printf("y=%d\n",y);
}
```

答案: y=4

143、#include void main () { int a[3][3]={{2,3,4},{5,6,8},{1,5,2}};

```
#include<stdio. h>
void main() {
    int a[3][3]={{2,3,4},{5,6,8},{1,5,2}};
    int *p=(int *)a;
    int s=0;
    while(p<(int *)a+9) s+= *p++;
    printf("%d\n",s);
}
```

执行下面程序的输出结果为()。

答案: 36

144、#include #define M6

```
#include<stdio.h>
#define M6
void main()
{
    inti, x=0
    inta (M)
    for(i=0;
    ={2, 5, 8, 4, 6, 9};
    i<M;i++)x +=a[i];
    printf("%d\n", x);
}
```

输出结果:

答案: 34

145、#include #define N 4 void main () { int a[N]; int i; a[0]=1;

```
#include<stdio.h>
#define N 4void main () {
    int a[N]; int i; a[0]=1;
    for (i=1; i<N; i++) a[i]=a[i-1]+i*i;
    printf ("%d\n", a[N-1]) ;
}
```

执行下面程序的输出结果为 (15)

146、#include #define NS

```
#include<stdio.h>
#define NS
void main(){
    inta[N]={3, 5, 8, 10, 15, 20, 32, 66};
    Intl, x;
    for(i=0;i<N/2;i++){
        x=a[i];a [i]=a[N - 1 - i] ;a [N - 1 - i]=x;
    }
    printf("%d\n", a [1]+a[3]);
```

答案: 47

147、#include #defineN5 void main () {

```
#include<stdio.h>
#defineN5
void main () {
    int a[N]
    int i; A[0]=1
    for (i=1; i<N; i++) a[i]=2*a[i-1]+1;
    printf ("%d\n", a[N-1]);
}
```

执行下面程序的输出结果为 ()

答案: 31

148、#include #void main () { inti, fl=1, f2;

```
include<stdio.h>
#void main(){
    inti, fl=1, f2;
    for(i=2;i<6;i++)
    f2=3*f1
    fl=f2
    printf("fZ=%d\n", f2);
}
```

答案: f2=81

149、#include struct Worker{char name[15];int age;float pay;

```
#include<stdio.h>
struct Worker{char name[15];int age;float pay;
void main(){
    struct Worker x={"wanghua", 48, 2500};
    struct Workery, *P;
    y=x;p=&x
    printf("%d\n", y.age+p -->age);
```

输出结果:

答案: 96

150、#include void main () { inta~25, b=18;

执行下面程序的输出结果为(

```
#include<stdio.h>
void main(){
    inta~25, b=18;
    if(a>b)printf("%S\n", "a>b");
    if(a==b)printf("%s\n", "a==b");
    if(a<b)printf("%s\n", "a<b");
```

答案: a>b

151、#include void main () { inta[8]={3, 8, 6, 5, 9, 2, 8, 5};

```
#include<Stdio.h>
void main(){
    inta[8]={3, 8, 6, 5, 9, 2, 8, 5};
    inti, s=0;
    for(i=0;i<8;i++)
        if(i%2)s+=a (i);
    printf("S=%d\n", s);
```

答案: s=20

152、#include void main () { inta (8) ={10, 8, 16, 15, 9, 21, 7, 16};

```
#include<stdio.h>
void main(){
    inta (8) ={10, 8, 16, 15, 9, 21, 7, 16};
    inti, x=a (0) ;
    for(i=1;i<8;i++)
        if(a (i) <x)x=a (i) ;
    printf("x=%d\n", x);
```

答案: x=7

153、#include void main () { intn=0, y=0;

```
#include<stdio.h>
void main(){
    intn=0, y=0;
    while(n<10){n++, y+=n
    printf("y=%d\n", y);
}
```

答案: y=55

154、#include Void main () {int n=10, y=0;

```
#include<stdio.h>
Void main () {
    int n=10, y=0;
    while (n>6) (y+=n; n--);
    printf ("y=%d\n",y);
}
```

执行下面程序的输出结果为 (y=34)。

155、#include void main () {inta (8) ={12, 15, 20, 8, 5, 9, 16, 10};

```
#include<stdio.h>
void main(<
    int a (8) ={12, 15, 20, 8, 5, 9, 16, 10};
    int i, j, x=a[0] +a (1) ;
    for(i=0;i<8;i++)
    for(j=i+1;j<8;j++)
    if(a[i] +a (j) >x)x=a (i) +a (j) ;
    printf("%d", x);
}
```

答案: 36

156、#include #define N 8

```
# include<stdio. h>
# define N 8
void main() {
    int a[N]={3,5,8,10,15,20,32,66};
    int i, x;
    for(i=0; i<N/2; i++) {
        x=a[i]; a[i]=a[N-1-i]; a[N-1-i]=x;
    }
    printf("%d\n", a[1]+a[3]);
}
```

答案: 47

157、#include #define N 5 void main () {

```
# include<stdio. h>
# define N 5
void main() {
    int a[N];
    int i;
    a[0]=1; a[1]=2;
    for(i=2; i<N; i++) a[i]=3 * a[i-2]+a[i-1]+1;
    printf("%d\n", a[N-1]);
}
```

答: 32

158、#include void main () { int x=12,s=0;

```
# include<stdio. h>
void main() {
    int x=12, s=0;
    do {s+=x; x-=3;} while(x>0);
    printf("s=%d\n", s);
}
```

答案: s=30

159、#include void main () { int a[8]={3,8,6,5,9,2,8,5};

```
# include<stdio. h>
void main() {
    int a[8]={3,8,6,5,9,2,8,5};
    int i, s=0;
    for(i=0; i<8; i++)
        if(a[i]>5) s+=a[i];
    printf("s=%d\n", s);
}
```

答: s=31

160、#include void main() { int a[8] = { 10, 8, 16, 15, 9, 21, 7, 16};

```
# include<stdio. h>
void main() {
    int a[8]={10,8,16,15,9,21,7,16};
    int i, x=a[0];
    for(i=1; i<8; i++)
        if(a[i]<x) x=a[i];
    printf("x=%d\n", x);
}
```

答: x=7

161、#include void main () { int i,f1=1,f2;

```
# include<stdio. h>
void main( ) {
    int i, f1=1, f2;
    for(i=2; i<5; i++) {
        f2=3 * f1;
        f1=f2;
    }
    printf("f2=%d\n", f2);
}
```

执行下面程序的输出结果为 (f2=27)

162、#include void main () { int x=5 ,y=8;

```
# include<stdio. h>
void main() {
    int x=5, y=8;
    if(x>=y) printf("%d %d\n", x, y);
    else printf("%d %d\n", y, x);
}
```

答: 8 5

163、#include void main () { int x=5,y=0;

```
# include<stdio. h>
void main() {
    int x=5, y=0;
    switch(2 * x+3) {
        case 4: y+=x; break;
        case 7: y+=2 * x+1; break;
        case 10: y+=3 * x; break;
        default: y+=x-1;
    }
    printf("y=%d\n", y);
}
```

答: y=4

164、#include void main () { int x=5,y=8,z=4;

```
#include<stdio.h>

void main( ) {
    int x=5,y=8,z=4;
    if(x>y) x=y;
    if(x>z) x=z;
}
```

执行下面程序的输出结果为 (4)

165、#include void main () { inti, s1=0, s2=0;

```
#include<stdio.h>

void main() {
    int i, s1=0, s2=0;
    for(i=1; i<10; i++)
        if(i%2) s1+=i;
        else s2+=i;
    printf("%d %d\n", s1, s2);
}
```

答: 25 20

166、#include void main () { intx=1,y;

```
#include<stdio.h>

void main() {
    int x=1,y;
    do {y=2 * x+1; x=y;} while(y<30);
    printf("y= %d\n",y);
}
```

答: y=31

167、#include void main () { int a=5, b=8, c=6, x;

```
#include<stdio.h>
void main () {
int a=5, b=8, c=6, x;
if (a>b) x=a; elsex=b;
if (c>x) x=c
printf ("%d\n", x);
}
```

执行下面程序的输出结果为 ()

答案: 8

```
168、#include void main () { int a[4][3]={ (3, 8, 6), {5, 9, 2},
{8, 5, 12}, {5, 13, 10}};
#include<stdio.h>
void main () {
int a[4][3]={ (3, 8, 6), {5, 9, 2}, {8, 5, 12}, {5, 13, 10}};
inti,j,s=0;
for (i=0; i<4; i++)
for (j=0; j<3; j++)
if (a[i][j]>8) s+=a[i][j];
printf ("s=%d\n", s);
}
```

执行下面程序的输出结果为 ()

答案: s=44

169、#include Void main () { int x=5,y=8,z=6,w=0;

```
#include<stdio.h>
Void main () {
int x=5,y=8,z=6,w=0;
if(x>y)w=x; else w=y;
if(z>w) w=z;
printf("w=%d\n", w);
}
```

执行下面程序的输出结果为 (w=8)。

170、#include void main () { int x=5, y=15, z=18;

```
#include<stdio.h>
void main () {
int x=5, y=15, z=18;
if (x>y&& x>z) printf ("%d\n", x) ;
else if (y>x&& y>z) printf ("%d\n", y) ;
else printf ("%d\n", z) ;
}
```

执行下面程序的输出结果为 (18)

171、#include void main () { int x=5, y=8, z=6, w=0;

```
#include<stdio.h>
void main(){
int x=5, y=8, z=6, w=0;
if(x>=y) w=x; else w=y;
if(w<z)w=z;
printf("w=%d\n", w);
}
```

答案: w=8

172、#include void main () { inta[6] ={36, 25, 48, 24, 55, 40};

```
#include<stdio.h>
void main(){
inta[6] ={36, 25, 48, 24, 55, 40};
int i=0, x=24;
while(a [i] !=x&& i<6)i++;
if(i<6)printf("%d\n", i);
}
```

输出结果:

答案: 3

173、#include void main () { inti, s=0;

```
#include<stdio.h>
void main(){
inti, s=0;
for(i=1;i<10;i++)
if(i%2==0)s+
printf("s=%d\n", s);
}
```

输出结果:

答案: s=20

174、#include Void main () {int a[8]={2,5,6,8,4,9,10,6};

```
#include<stdio.h>
Void main () {
int a[8]={2,5,6,8,4,9,10,6};
int i,x=a[0]+a[1];
for (i=1; i<7; i++) {
int y=a[i]+a[i+1];
if(y>x)x=y;
}
printf("0/<d\n", x);
}
```

执行下面程序的输出结果为 (19)。

175、#include void main () {int i=0, s=0;

```
#include<stdio.h>
void main () {
int i=0, s=0; while (s<25) {i+=2; s+=i*i; }
printf ("s=%d\n", s) ;
}
```

执行下面程序的输出结果为 (s=563)

176、#include void main () {inti=0, s=0;

```
#include<stdio.h>
void SB(char ch){
switch(ch){
case'A':printf("1");break;
case'B':printf("2");break;
default:printf("3");break;
}
```

```
void main(){Char al='a', aZ ='B', a3='A';
SB(al);SB(aZ);SB(a3);
}
```

输出结果:

答案: 321

178、#include voidmain () { int n=2, y=0; while (n<10) {y+=n; n+=2; }

```
#include<stdio.h>
```

```
voidmain () {
int n=2, y=0; while (n<10) {y+=n; n+=2; }
```

```
printf("y=%d\n", y);
}
```

执行下面程序的输出结果为 ()。

答案: y=20

179、**#include void main ()**。

执行下面程序的输出结果为 ()。

```
#include<stdio.h>
```

```
void main ( )。
```

```
int i=0,s=0;
```

```
while (s<30) (i+=2; s+=i*i);
```

```
printf (" i=%d\n " ,i)。
```

```
)
```

答: i=6

180、**#includevoid main () {char ch;**

```
int a,b,c;
```

```
scanf("%c",&ch);
```

```
scanf("%d %d",&a,&b);
```

```
switch(ch) {case '+': c=a+b;Break;
```

```
case '-': c=a-b;Break;
```

```
case '*': c=a*b;Break;
```

```
case '/': c=a/b;Break;
```

```
default: printf("Error!\n");
```

```
} printf("%d%c%d=%d\n",a,ch,b,c);
```

```
}
```

执行下面程序时,假定从键盘上输入的字符为*,输入的两个整数为3和5,则程序输出结果为 (A.3*5=15)。

181、**#includevoid main () {int a=25,b=18;**

```
if(a>b) printf("%s\n","a>b");
```

```
if(a==b) printf("%s\n","a==b");
```

```
if(a
```

执行下面程序的输出结果为 (A.a>b)。

182、**#includevoid main () {int a=5,b=8,c;**

```
if(3*a>2*b) c=2*a+3*b;
```

```
else if(a==b) c=a*a+b*b;
```

```
else c=a*b+3;
```

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

```
}
```

执行下面程序的输出结果为 (A.c=43)。

183、**#includevoid main () {int a=5,b=8,c=6,d=15,x,y;**

```
if(a>=b) x=a;
```

```
else x=b;
```

```
if(c>=d) y=c;
```

```
else y=d;
```

```
if(x>=y) printf("%d\n",x);
```

```
else printf("%d\n",y);
```

```
}
```

执行下面程序的输出结果为 (A.15)。

184、**#includevoid main () {int a[8]={10,8,12,15,9,27,32,16};**

```
int i,x=a[0];
```

```
for(i=1;
```

```
i<8;
```

```
i++) if(a[i]>x) x=a[i];
```

```
printf("x=%d\n",x);
```

```
}
```

执行下面程序的输出结果为 (A.x=32)。

185、**#includevoid main () {int a[8]={10,8,16,15,9,21,7,16};**

```
int i,x=a[0];
```

```
for(i=1;
```

```
i<8;
```

```
i++) if(a[i]
```

执行下面程序的输出结果为 (A.x=7)。

186、**#includevoid main () {int a[8]={10,8,6,15,9,21,8,16};**

```
int i,x=a[0];
```

```
for(i=1;
```

```
i<8;
```

```
i++) if(a[i]>x) x=a[i];
```

```
printf("x=%d\n",x);
```

```
}
```

执行下面程序的输出结果为 (A.x=21)。

187、**#includevoid main () {int a[8]={3,8,6,5,9,2,8,5};**

```
int i,s=0;
```

```
for(i=0;
```

```
i<8;
```

```
i++) if(a[i]>5) s+=a[i];
```

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

```
}
```

执行下面程序的输出结果为 (A.s=31)。

188、**#includevoid main () {int i, s1=0, s2=0;**

```
for(i=1;
```

```
i<10;
```

```
i++) if(i%2) s1+=i;
```

```
else s2+=i;
```

```
printf("%d %d\n",s1,s2);
```

```
}
```

执行下面程序的输出结果为 (A.25 20)。

189、**#includevoid main () {int i, s=0;**

```
for(i=1;
```

```
s<20;
```

```
i+=2) s+=i;
```

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

```
}
```

执行下面程序的输出结果为 (A.s=25)。

190、**#includevoid main () {int i,f,f1=1,f2=2;**

```
for(i=3;
```

```
i<=6;
```

```
i++) {f=f1+f2;
```

```
f1=f2;
```

```
f2=f;
```

```
} printf("f=%d\n",f);
```

```
}
```

执行下面程序的输出结果为 (A.f=13)。

191、**#includevoid main () {int i,s=0;**

```
for(i=1;
```

```
i<=6;
```

```
i++) s+=i*i;
```

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

```
}
```

执行下面程序的输出结果为 (A.s=91)。

192、**#includevoid main () {int i,s=0;**

```
for(i=16;
```

```
i>0;
```

```
i-=2) if(i%3==0 || i%4==0) s+=i;
```

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

```
}
```

执行下面程序的输出结果为 (A.s=46)。

193、**#includevoid main () {int i,s=0;**

```
for(i=1;
```

```
i<10;
```

```
i++) {s+=i*(i+1);
```

```
if(s>40) break;
```

```
} printf("s=%d\n",s);
```

```
}
```

执行下面程序的输出结果为 (A.s=70)。

194、**#includevoid main () {int i,s=0;**

```
for(i=1;
```

```
i<10;
```

```
i++) {if(i%2) continue;
```

```
s+=i*(i-1);
```

```
} printf("s=%d\n",s);
```

```
}
```

执行下面程序的输出结果为 (A.s=100)。

195、**#includevoid main () {int i,x,y,s=0;**

```
scanf("%d %d",&x,&y);
```

```
if(x>y) {i=x;
```

```
x=y;
```

```
y=i;
```

```
} for(i=x;
```

```
i<=y;
```

```
i++) if(i%2==0) s+=i*i;
```

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

```
}
```

下面程序的功能是求出从键盘上输入的两个整数之间所有 (A.偶数) 的平方和。

196、**#includevoid main () {int i=0, s=0;**

```
while(s<25) {i+=2;
```

```
s+=i*i;
```

```
} printf("s=%d\n",s);
```

```
}
```

执行下面程序的输出结果为 (A.s=56)。

197、**#includevoid main () {int x=5,y=0;**

```
switch(2*x-3) {case 4: y+=x;Break;
```

```
case 7: y+=2*x+1;Break;
```

```
case 10: y+=3*x;Break;
```

```
} printf("y=%d\n",y);
```

执行下面程序的输出结果为 (A.v=11)。

```
198、#include void main () {int x=5,y=8,z=12;
if(x>y && x>z) printf("%d\n",x);
else if(y>x && y>z) printf("%d\n",y);
else printf("%d\n",z);
}
```

执行下面程序的输出结果为 (A.12)。

```
199、#include void main () {int x=5,y=8,z=14,w;
if(x
```

执行下面程序的输出结果为 (A.14 8 5)。

```
200、#include void main () {int x=5,y=8,z=4;
if(x>y) x=y;
if(x>z) x=z;
printf("%d\n",x);
}
```

执行下面程序的输出结果为 (A.4)。

```
201、#include void main () {int x=5,y=8,z=6,w=0;
if(x>=y) w=x;
else w=y;
if(w
```

执行下面程序的输出结果为 (A.w=8)。

```
202、#include void main () {int x=5,y=8;
if(x>=y) printf("%d %d\n",x,y);
else printf("%d %d\n",y,x);
}
```

执行下面程序的输出结果为 (A.8 5)。

```
203、#include void main () {int x=9,y=0;
switch(x%3) {case 0: y+=x;
case 1: y+=2*x+1;
case 2: y+=3*x;Break;
default: y+=x-1;
} printf("y=%d\n",y);
}
```

执行下面程序的输出结果为 (A.v=55)。

```
204、include<stdio. h>void main(
```

```
#include<stdio. h>
void main( ) {
int a[8]={10,8,12,15,9,27,8,16};
int i,x=a[0];
for(i=1;i<8;i++)
if(a[i]>x) x=a[i];
printf("x=%d\n",x);
}
```

执行下面程序的输出结果为 (x=27)

```
205、include<stdio.h> void main () { int x=5
```

```
24. #include<stdio. h>
void main()
{
int x=5;
switch(2 * x-1){
case 4: printf("%d", x); break;
case 7: printf("%d", 2 * x); break;
case 10: printf("%d", 3 * x); break;
default: printf("%s", "default");
}
}
```

输出结果:

答案: default

```
206、include<stdio.h> void main () {int f1=1,f2,i
```

```
25. #include<stdio. h>
void main() {
int f1 = 1, f2, i;
printf("%d ", f1);
for(i = 1; i < 4; i++) {
f2 = 3 * f1 + 1;
printf("%d ", f2);
f1 = f2;
}
}
```

输出结果:

答案: 1 4 13 40

```
207、include<stdio.h>defineN6
```

```
2. #include<stdio. h>
```

```
#define N 6
void main()
{
int i,a[N]={2,5,8,10,15,21};
for(i=0; i<N; i++)
if(a[i]%5) printf("%d ", a[i]);
printf("\n");
}
```

答案: 2 8 21

```
208、include<stdio.h>intWF(intx,inty){
```

```
#include<stdio. h>
int WF(int x, int y){
x=x+y;
Y+=x;
return x+ y;
}
void main(){
int x = 3, y=5;
printf ("%d\n", WF(x, y));
}
```

答案: 21

```
209、include<stdio.h>void main () { int a,b,y
```

```
4. #include<stdio. h>
void main()
{
int a,b;
for(a=2,b=3; b<20;) {
printf("%d %d ", a, b);
a=a+b;
b=a+b;
}
printf("%d %d\n", a, b);
}
```

答案: 2 3 5 8 13 21

210、`include<stdio.h>void main () { int i;`

```
3. #include<stdio.h>
#include<string.h>
```

```
void main() {
    int i;
    unsigned int len;
    char * a[5]={"student","worker","cadre","soldier","zzeasan123"};
    len=strlen(a[0]);
    for(i=1; i<5; i++)
        if(strlen(a[i])>len) len=strlen(a[i]);
    printf("%d\n",len);
}
```

答案: 10

211、`include<stdio.h>void main () { int i,j,k=0;`

```
#include<stdio.h>
void main(){
    int i,j,k=0;
    for(i=0;i<5;i++)
        for(j=i; j<5;j++)k++
    printf("%d/n", k);
}
```

答案: 15

212、`include<stdio.h>voidLE(inta,intb){`

```
5. #include<stdio.h>
void LE(int * a, int * b) {
    int x= * a;
    * a= * b; * b=x;
}
void main() {
    int x=15, y=26;
    printf("%d %d\n",x,y);
    LE(&x,&y);
    printf("%d %d\n",x,y);
}
```

答案: 15 26
26 15

213、`include<stdio.h>voidmain () {int i,s=0;`

1. `#include<stdio.h>`

```
void main() {
    int i,s=0;
    for(i=1;i<6;i++) s+=i * i;
    printf("s= %d\n",s);
}
```

答案: s=55

214、`include<stdio.h>[intWF(intx,inty){`

```
#include<stdio.h>
int WF(int x, int y){
    x=x+y;
    Y+=x;
    return x+ y;
}
void main(){
    int x= 3, y=5;
    printf ("%d\n" , WF(x, y));
}
```

答案: 21

215、`includeconstintM=20`

```
#includeConst int M=20;
void main()
{
    int i,c2,c3,c5;C2=c3=c5=0;
    for(i=1; i<=M; i++) {
        if(i%2==0) c2++;
        if(i%3==0) c3++;
        if(i%5==0) c5++;
    }
    printf("%d %d %d\n",c2,c3,c5);
}
```

答案: 10 6 4

216、`includeinta[10]={4,5,6,15,20,13,12,7,8,9}`

```
#include
int a[10]={4,5,6,15,20,13,12,7,8,9};
void main()
{
    int i,s[4]=0;
    for(i=0; i<10; i++) {
        switch(a%4) {Case 0: s[0]+=a;break;Case 1: s[1]+=a;break;Case 2:
s[2]+=a;break;Default: s[3]+=a;break;
        }
    }
    for(i=0; i<4; i++) printf("%d ",s);
}
```

答案: 4 4

```
217、includevoidfun(charss[])
#include
void fun(char ss[]);
void main() {Char s[15]="0123456789";
fun(s);
printf("%s\n",s);
}
void fun(char ss[]) {
int i, n=strlen(ss) ;
for(i=0; i<n/2; i++) {Char c=ss;
ss=ss[n-1-i];
ss[n-1-i]=c;
}
```

答案: 9876543210

218、`includevoidmain ()`

```
#include
void main()
{
    int x=5;
    switch(2*x-3) {Case 4: printf("%d ",x);Case 7: printf("%d
",2*x+1);Case 10: printf("%d ",3*x-1); break;Default:
printf("%s","default\n");
}
}
```

答案: 11 14

219、`includevoidmain ()`

```
#include
void main()
{
    int i,s;
    for(i=1,s=0;i<10;i++) {
        if(i%3==0) continue;
        s+=i;
    }
    printf("%d\n",s);
}
```

答案: 27

220、`includevoidmain () {`

```
#include
void main() {
    int i, s1=0, s2=0;
    for(i=0;i<10;i++)
        if(i%2) s1+=i; else s2+=i;
    printf("%d %d\n",s1,s2);
}
```

答案: 25 20

221、`includevoidmain () {`

```
#include
void main() {
```

```
int n=6,y=1;
while(n) y*=n--;
printf("y=%d\n",y);
}
```

答案: y=720

222、`includevoidmain () {`

```
#include
void main() {
int a[8]={12,39,26,41,55,63,72,40};
int i, i1=0, i2=0;
for(i=0;i<8;i++)
if(a%2==1) i1++; else i2++;
printf("%d %d\n",i1,i2);
}
```

答案: 4 4

223、`includevoidmain () {`

```
#include
void main() {Char a[]="abcdbfbgacd";
int i1=0, i2=0, i=0;
while(a) {
if(a=='a') i1++;
if(a=='b') i2++;
i++;
}
printf("%d %d %d\n",i1,i2,i);
}
```

答案: 2 3 11

224、`includevoidmain () {`

```
#include
void main() {
int a[3][4]={{1,2,7,8},{5,6,10,6},{9,12,3,4}};
int m=a[0][0];
int i,j;
for(i=0; i<3; i++)
for(j=0; j<4; j++) if(a[j]>m) m=a[j];
printf("%d\n",m);
}
```

答案: 12

225、`int xk3(int n){int i,s=1,`

```
for(i=1;
i<=n;
i++)s*=i;
return s;
}
```

假定使用 `xk3(4)`调用下面函数, 则返回值为(24)。

226、`intmain () {inti,sum=0;`

```
intmain(){inti,sum=0;
for(i=0;i<=100;i++)sum=sum+i;
printf("sum=%d\n",sum);
return0;
```

} 以下程序的输出是 (C.5050)

227、`void main () { int a[3][3]=1(3,5,7),(9,11,13), (6,8,20);`
答案: k=>

228、当使用 `xkZC` 调用此函数时, 返回值为()

当使用 `xkZC` 调用此函数时, 返回值为()。

```
int xk2(int a, int b, int c){
if(a>=b&& a>=c)return a,
if(b>=a 且 b>=c)return b,
return c;
```

答案: 7

229、假定使用 `xxk3(5)`调用下面函数, 则返回值为()。`int xxk3(int`

```
n){
int xxk3(int n){
int i,s=1;
for(i=1;i<=n;i++)s*=i;
return s;
}
```

答案: 120

230、假定使用 `xxk3 (4)`调用下面函数, 则返回值为 ()

```
int xxk3(int n) {
int i, s=1,
for(i=1; i<=n; i++) s*=i;
return s;
}
```

答案: 24

函数功能(26)--电大资源网: <http://www.dda123.cn/> (微信搜: 905080280)

- 1、`#include<math.h>`
- 2、`#include<stdio.h> int SA(int a, int b) {...`
- 3、`double SF(double x, int n) { //n 为正整数...`
- 4、`int * LI(int n)`
- 5、`int Count(struct IntNode *f)`
- 6、`int Count(struct IntNode *f)`
- 7、`int FF(struct IntNode*f)`
- 8、`int FH ()`
- 9、`int FindMax(struct IntNode *f) //f 为一...`
- 10、`int fun(int m, int n) {`
- 11、`int i=2; if(x==2||x==3) return 1...`
- 12、`int LK(double a[], int n) {`
- 13、`int SC(int a,int b,int c)`
- 14、`int SG(int x) { //x 为大于等于...`
- 15、`int WB(int a[], int n, int x) {`

16、`int WC(int a[], int n, int k) { int i, c=0...`

17、`void Output(struct IntNode *f) //f 为单...`

18、`void QA(struct Worker a[], int n) {`

19、`void QA(struct Worker a[], int n) {int i;...`

20、`void xw1(char* fname) {`

21、`void xw1(char* fname) {`

22、`voidQA(structWorkera[],intn){`

23、假定结点的结构类型为 `IntNode`, 它的数值域为 `d...`

24、假定结点的结构类型用 `IntNode` 表示, 它的数值...

25、假定结构类型 `struct Worker` 的定义体为: ...

26、假定使用 `SF (20)`调用下面函数, 则得到的输出结果...

1、`#include<math.h>`

`#include<math.h>`

`int SG (int x) { //x 为大于等于 2 的整数`

`int a= (int) sqrt (x) ; //sqrt (x) 取 x 的平方根`

`int i=2;`

`while (i<=a) {`

`if (x%i==0) break;`

`i++;`

`}`

`if (i<=a) return 0;else return 1;`

函数功能:

答案: 判断 x 是否为一个素数, 若是则返回

2、`#include<stdio.h> int SA(int a, int b) {`

`# include<stdio. h>`

`int SA(int a, int b) {`

`if(a>b) return 1;`

`else if(a==b) return 0;`

`else return -1;`

`}`

函数功能:

答案: 比较两个整数 a 和 b 的大小, 若 a>b 则返回 1, 若 a=b 则

返回 0, 若 a<b 则返回-1。

3、`double SF(double x, int n) { //n 为正整数 Double SF(double x,`

`int n) { //n 为正整数`

`double p=1,s=1;`

`int i;`

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

`p*=x;`

`s+=p;`

`}`

`return s;`

`}`

答案: 计算出 $1+x+x^2+\dots+x^n$ 的值并返回。

4、`int * LI(int n)`

`int*LI (int n) {`

```
int i;
int*a=malloc (n*sizeof (int) );
for (i=0; i<n; i++) scanf ("%d", a+i) ;
return a;
```

函数功能:

答案: 首先建立一个动态整型数组 a[n], 接着从键盘上输入 n 个整数到 a[n]中, 最后返回数组 a[n]的首地址。

5、int Count(struct IntNode *f)

```
int Count(struct IntNode *f)
{ //f 为指向一个单链表的表头指针
  int c=0;
  while(f) {
    c++;
    f=f->next;
  }
  return c;
}
```

假定 struct IntNode 的类型定义为:

```
struct IntNode {int data; IntNode* next;};
```

答案: 统计出以表头指针为 f 的链表中结点的个数。

6、int Count(struct IntNode *f)

```
int Count(struct IntNode *f) { //f 为指向一个单链表的表头指针
```

```
  int c=0;
  while(f) {
    c++;
    f=f->next;
  }
  return c;
}
```

假定 struct IntNode 的类型定义为:

```
struct IntNode {int data; IntNode* next;};
```

答案: Count 函数可以统计单链表的结点数, 其表头指针为 f

7、int FF(struct IntNode*f)

```
int FF (struct IntNode*f) //f 为一个单链表的表头指针
```

```
{
  int c=0;
  if(f==NULL)return 0;
  while(f){c++;f=f->next;
}
```

```
return c;
```

函数功能:

答案: 求出并返回由 f 所指向的单链表中结点的个数。

8、int FH ()

```
int FH() {
  int x,y=0;
  scanf("%d", &x);
  while(x>0) (y+=x; scanf("%d", &x));
  return y;
}
```

答案: 求出从键盘上输入的一批整数的总和, 以小于等于 0 作为

结束键盘输入的标志。

9、int FindMax(struct IntNode *f) //f 为一个单链表的表头指针

```
int FindMax(struct IntNode *f) //f 为一个单链表的表头指针
```

```
{ int x;
  if(!f) {printf("单链表为空\n"),exit(1);}
  x=f->data;
  f=f->next;
  while(f) {
    if(f->data>x) x=f->data;
    f=f->next; }
  return x; }
```

假定 struct IntNode 的结点类型定义为:

```
struct IntNode { int data; struct IntNode* next;};
```

函数功能: 求出并返回由 f 所指向的单链表中所有结点的最大值。

10、int fun(int m, int n) {

```
int fun(int m, int n) {
  int c=0;
  static int b=2;
  if(m<b || n<b) return m*n;
  else if(m%b==0 && n%b==0) {c=b; return
c*fun(m/b,n/b);}
  else {b++; return fun(m,n);}
}
```

答案: 一个递归函数过程, 求出两个自然数 m 和 n 的最小公倍数。

11、int i=2; if(x==2||x==3) return 1

```
int i=2;
if(x==2||x==3) return 1;
while(i*i<=x){
  if(x%i==0) break;
  i++;
}
if (i * i<=x)return 0;else return 1;
```

函数功能:

答案: 判断 x 是否为一个素数, 若是则返回 1, 否则返回 0。

12、int LK(double a[], int n) {

```
int LK(double a[], int n) {
  double s=0;
  int i,m=0;
  for(i=0;i<n;i++) s+=a[i];
  s/=n;
  for(i=0;i<n;i++)
    if(a[i]>=s) m++;
  return m;
}
```

答案: 求出并返回数组 a 的 n 个元素中大于等于平均值的元素个数。

13、int SC(int a,int b,int c)

```
int SC(int a, int b, int c) {
  if(a>=b && a>=c) return a;
  if(b>=a && b>=c) return b;
```

```
return c;
```

```
}
```

函数功能:

答案: 求出 a, b, c 三个数中的最大值并返回。

14、int SG(int x) { //x 为大于等于 2 的整数

```
int SG(int x) { //x 为大于等于 2 的整数
```

```
  int i=2;
  while(i*i<=x) {
    if(x%i==0) break;
    i++;
  }
  if(i*i<=x) return 0; else return 1;
```

答案: 判断 x 是否为一个质数 (素数), 若是则返回 1, 否则返回 0。

15、int WB(int a[], int n, int x) {

```
int WB(int a[], int n, int x) {
  for(int i=0;i<n;i++)
    if(a[i]==x) return 1;
  return 0;
}
```

答案: 从数组 a[n]中顺序查找值为 x 的元素, 若查找成功则返回 1, 否则返回 0。

16、int WC(int a[], int n, int k) { int i, c=0;

```
int WC(int a[], int n, int k) {
  int i, c=0;
  for(i=0; i<n; i++)
    if(a[i]>=k) c++;
  return c; }
```

函数功能: 统计并返回一维整型数组 a[n]中大于等于 k 的值的个数。

17、void Output(struct IntNode *f) //f 为单链表的表头指针

```
void Output(struct IntNode *f) //f 为单链表的表头指针
{ if(!f) return;
  while(f) {
    printf("%d ",f->data);
    f=f->next; }
  printf("\n"); }
```

假定 struct IntNode 的类型定义为:

```
struct IntNode { int data; struct IntNode* next;};
```

函数功能: 遍历并输出由 f 所指向的单链表中每个结点的值。

18、void QA(struct Worker a[], int n) {

```
void QA(struct Worker a[], int n) {
  int i;
  for(i=1; i<n; i++)
    scanf("%s %d %f",&a[i].name,&a[i].age,&a[i].pay);
}
```

假定结构类型 struct Worker 的定义如下:

```
struct Worker { char name[15]; int age; float pay;};
```

答案: 从键盘输入 n 个 Worker 类型的记录到一维数组 a 中。

19、void QA(struct Worker a[], int n) {int i;

```
void QA(struct Worker a[], int n)
```

```
{int i;
for(i=0; i<n; i++)
scanf("%s%d%f", a[i].name,&a[i].age,&a[i].pay);}
```

假定结构类型 struct Worker 的定义如下:

```
struct Worker{char name[15];int age;float pay;};
```

函数功能:从键盘上为具有 struct Worker 类型的数组 a[n]输入 n 个记录。

20、void xw1(char* fname) {

```
void xw1(char* fname) {
FILE* fout=fopen(fname,"w");
char a[20];
printf("每个字符串长度小于 20, 字符串 end 作为结束标志\n");
while(1) {
scanf("%s",a);
if(strcmp(a,"end")==0) break;
fputs(a,fout);
fputc('\n',fout);
}
fclose(fout);
}
```

答案: 把从键盘上输入的若干个字符串保存到由 fname 参数所指定的文本文件中。

21、void xw1(char* fname) {

```
void xw1(char* fname) {
FILE* fout=fopen(fname,w);Char a[20];
printf(每个字符串长度小于 20, 字符串 end 作为结束标志\n);
while(1) {
scanf(%s,a);
if(strcmp(a,end)==0) break;
fputs(a,fout);
fputc('\n',fout);
}
fclose(fout);
}
```

答案: xw1 函数可以输入若干字符串保存到 fname 所指定的文件中, 直至遇到字符串 end

22、void QA(struct Workera[],intn){

```
void QA(struct Worker a[], int n) {
int i;
for(i=1; i<=n; i++)
scanf(%s %d %f,&A.name,&A.age,&A.pay);
}
```

假定结构类型 struct Worker 的定义如下:

```
struct Worker { char name[15]; int age; float pay;};
```

答: 从键盘输入 n 个 worker 类型的记录到一组数组 a 中。

23、假定结点的结构类型为 IntNode, 它的数值域为 data, 链接指针域为 next, 请指出下面函数的功能。

```
void QC (struct IntNode*f) { //f 为一个单链表的表头指针
while (f) {
```

```
printf ("%d", f->data) ;
```

```
f=f->next;
```

函数功能:

答案: 遍历输出由表头指针 f 所指向的单链表中每个结点的 data 域的值。

24、假定结点的结构类型用 IntNode 表示, 它的数值域为 data, 链接指针域为 next, 请根据下面函数定义写出相应的功能。

假定结点的结构类型用 IntNode 表示, 它的数值域为 data, 链接指针域为 next, 请根据下面函数定义写出相应的功能。

int FF(struct IntNode*f)//f 为指向一个单链表的表头指针

```
{
Int n=0
if(f==NULL)returno;
while(f){n++;f=f->next;}
returnnll;
}
```

函数功能:

答案: 求出并返回由 f 所指向的单链表中所含结点的个数。

25、假定结构类型 struct Worker 的定义体为:

```
{char name[15];int age; float pay;}
```

```
void QA (struct Worker a[], int n)
```

```
int i;
```

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

```
scanf ("%s%d%f", a[i].name, &a[i].age, &a[i].pay);
```

函数功能:
答案: 从键盘上为具有 struct Worker 类型的数组 a[n]输入 n 个记录。

26、假定使用 SF (20) 调用下面函数, 则得到的输出结果为

假定使用 SF (20) 调用下面函数, 则得到的输出结果为 ()

```
void SF (int x) {
int i=2; while (x>=i) {
if (x%i==0) {printf ("%d", i) ;x/=i;}
i++;
printf ("%d\n", x) ;
```

答案: 2 5 2