

- 1 C语言程序的基本单位是\_\_\_\_\_ A) 程序行 B) 语句 C) 函数 D) 字符 C
- 2 C语言程序的三种基本结构是\_\_\_\_\_ A A
  - A、顺序结构, 选择结构, 循环结构 B、递归结构, 循环结构, 转移结构
  - C、嵌套结构, 递归结构, 顺序结构 D、循环结构, 转移结构, 顺序结构
- 3 C语言规定, 程序中各函数之间 A
  - A) 既允许直接递归调用也允许间接递归调用
  - B) 不允许直接递归调用也不允许间接递归调用
  - C) 允许直接递归调用不允许间接递归调用
  - D) 不允许直接递归调用允许间接递归调用
- 4 C语言中可处理的文件类型是( ) B
  - A) 文本文件和数据文件 B) 文本文件和二进制文件
  - C) 数据文件和二进制文件 D) 数据代码文件
- 5 C语言可执行程序的开始执行点是( ) C
  - A) 程序中第一条可执行语句 B) 程序中第一个函数
  - C) 程序中的main函数 D) 包含文件中的第一个函数
- 6 不是C语言提供的合法的数据类型关键字是 A) double B) short C) integer D) char C
- 7 C语言中, 运算对象必须是整型数的运算符是 A) % B) \ C) %和\ D) \* \* A
- 8 C语言中函数返回值的类型是由( ) 决定。 D
  - A) return语句中的表达式类型 B) 调用函数的主调函数类型
  - C) 调用函数时临时 D) 定义函数时所指定的函数类型
- 9 C语言中数组名作为参数传递给函数, 作为实在参数的数组名被处理为\_\_\_\_\_。 D
  - A. 该数组的长度 B. 该数组的元素个数 C. 该数组中各元素的值 D. 该数组的首地址
- 10 C语言中数组下标的下限是\_\_\_\_\_ A、1 B、0 C、视具体情况 D、无固定下限 B
- 11 C语言中提供的合法关键字是\_\_\_\_\_ A、swith B、cher C、cese D、default D
- 12 C语言中文件的存取方式是\_\_\_\_\_。 C
  - A. 顺序存取 B. 随机存取 C. 顺序存取、随机存取均可 D. 顺序存取、随机存取均不可
- 13 C语言中最简单的数据类型包括 B
  - A) 整型、实型、逻辑型 B) 整型、实型、字符型
  - C) 整型、字符型、逻辑型 D) 整型、实型、逻辑型、字符型
- 14 sizeof(float)是( ) C
  - A. 一种函数调用 B. 一个不合法的表示形式 C. 一个整型表达式 D. 一个浮点表达式
- 15 x、y、z被定义为int型变量, 若从键盘给x、y、z输入数据, 正确的输入语句是 B
  - A) input x,y,z; B) scanf("%d%d%d",&x,&y,&z);
  - C) scanf("%d%d%d",x,y,z); D) read("%d%d%d",&x,&y,&z);
- 16 表达式:10!=9的值是 A) true B) 非零值 C) 0 D) 1 D
- 17 表示关系 $x \leq y \leq z$ 的c语言表达式为 A
  - A) (X<=Y)&&(Y<=Z) B) (X<=Y)AND(Y<=Z) C) (X<=Y<=Z) D) (X<=Y)&(Y<=Z)
- 18 程序片段:在TC20中, int i=65536; printf("%d\n",i);的输出结果是\_\_\_\_\_  B
  - A) 65536 B) 0 C) 有语法错误, 无输出结果 D) -1
- 19 当调用函数时, 实参是一个数组名, 则向函数传送的是\_\_\_\_\_ B
  - A. 数组的长度 B. 数组的首地址 C. 数组每一个元素的地址 D. 数组每个元素中的值
- 20 对嵌套子程序调用说法正确的是\_\_\_\_\_ 
  - A. 外层子程序可以调用所有的内层子程序
  - B. 内层子程序只可以调用包含本身的外层子程序, 不可以隔层调用
  - C. 外分程序必须能完全套住内分程序
  - D. 以上说法均不正确
- 21 对于基类型相同的两个指针变量之间, 不能进行的运算是 A) < B) = C) + D) - C
- 22 C语言中, 合法的长整型常数是 A) '\t' B) "A" C) 65 D) A  A
- 23 假定int类型变量占用两个字节, 其有定义:int x[10]={0,2,4};, 则数组x在内存中所占字节数是 D
  - A) 3 B) 6 C) 10 D) 20
- 24 假定有以下变量定义: int k=7 ,x=12; 则能使值为3的表达式是 D
  - A x%=(k%=5) B x%=(k-k%5) C x%=k-k%5 D (x%=k)-(k%=5)
- 25 请读程序片段(字符串内没有空格): printf("%d\n",strlen("ATS\n012\1\"));  C
  - 则输出结果是 A) 11 B) 10 C) 9 D) 8
- 26 请选出合法的C语言赋值语句 A) a=b=58 B) i++; C) a=58,b=58 D) k=int(a+b); B
- 27 若a为int类型, 且其值为3, 则执行完表达式a+=a-a\*a后, a的值是 A. -3 B. 9 C. -12 D. 6 C
- 28 若int类型占两个字节, 则以下语句输出为\_\_\_\_\_  D
 

```
int k=-1;
printf("%d,u\n",k,k);
```

  - A、-1,-1 B、-1,65536 C、-1,32768 D、-1,65535
- 29 若变量a是int类型, 并执行了语句:a= 'A'+1.6; , 则正确的叙述是 D
  - A) a的值是字符C B) a的值是浮点型
  - C) 不允许字符型和浮点型相加 D) a的值是字符 'A' 的ASCII值加上1。

- 30 若变量已正确说明为float类型,要通过语句scanf("%f %f %f ", &a, &b, &c);给a赋予10.0, b赋予22.0, c赋予33.0, 不正确的输入形式是:  
A) 10<回车> B) 10.0, 22.0, 33.0<回车> C) 10.0<回车> D) 10 22<回车> 22<回车> 22.0 33.0<回车> 33<回车> B
- 31 若要求在if后一对圆括号中表示a不等于0的关系, 则能正确表示这一关系的表达式为  
A) a<>0 B) !a C) a=0 D) a B
- 32 若已定义x和y为double类型, 则表达式:x=1, y=x+3/2的值是 A) 1 B) 2 C) 2.0 D) 2.5 C
- 33 若有定义:int a=8, b=5, c;, 执行语句c=a/b+0.4;后, c的值为 A) 1.4 B) 1 C) 2.0 D) 2 B
- 34 若有定义:int x, y; char a, b, c; 并有以下输入数据(此处< CR> 代表换行符, /u代表空格):  
1u2<CR>AuBuC  
则能给x赋整数1, 给y赋整数2, 给a赋字符A, 给b赋字符B, 给c赋字符C的正确程序段是  
A) scanf("x=%d y=%d", &x, &y); a=getchar(); b=getchar(); c=getchar();  
B) scanf("%d %d", &x, &y); a=getchar(); b=getchar(); c=getchar();  
C) scanf("%d%d%c%c%c", &x, &y, &a, &b, &c); D
- 35 若有以下定义和语句:  
char s[10]; s="abcd"; printf("%s\n", s);  
则结果是(以下u代表空格) A) 输出abcd B) 输出a C) 输出abcduuuuu D) 编译不通过 D
- 36 若有以下程序段, 则执行后, c3中的值是 A) 0 B) 0.5 C) 1 D) 2  
int c1=1, c2=2, c3;  
c3=1.0/c2\*c1; A
- 37 若有以下定义: 则表达式a\*b+d-c值的类型为 A) float B) int C) char D) double  
char a; int b;  
float c; double d; D
- 38 若有以下定义和语句 则输出结果是:  
char c1='b', c2='e';  
printf("%d, %c\n", c2-c1, c2-'a'+"A");  
A) 2, M B) 3, E C) 2, E D) 输出项与对应的格式控制不一致, 输出结果不确定 B
- 39 若有以下函数调用语句: fun(a+b, (x, y), fun(n+k, d, (a, b))); 在此函数调用语句中实参的个数是  
A) 3 B) 4 C) 5 D) 6 A
- 40 若执行下面的程序时从键盘上输入3和4, 则输出是\_\_\_\_\_  
main()  
{ int a, b, s;  
scanf("%d %d", &a, &b);  
s=a;  
if(a<b)s=b;  
s=s\*s;  
printf("%d\n", s);} A) 14 B) 16 C) 18 D) 20 B
- 41 若执行下面的程序时从键盘上输入5, 则输出是 A) 7 B) 6 C) 5 D) 4  
main()  
{ int x;  
scanf("%d", &x);  
if(x++>5) printf("%d\n", x);  
else printf("%d\n", x--);} B
- 42 若执行以下程序时从键盘上输入9, 则输出结果是 A) 11 B) 10 C) 9 D) 8  
main()  
{ int n;  
scanf("%d", &n);  
if(n++<10) printf("%d\n", n);  
else printf("%d\n", n--);} B
- 43 设 a、b、c、d、m、n均为 int型变量, 且 a=5、b=6、c=7、d=8、m=2、n=2, 则逻辑表达式  
(m=a>b)&&(n=c>d) 运算后, n的值为\_\_\_\_\_  
A) 0 B) 1 C) 2 D) 3 C
- 44 设 int a=12, 则执行完语句 a+=a-=a\*a后, a的值是 A) 552 B) 264 C) 144 D) -264 D
- 45 下面程序段的运行结果是  
int a=1, b=2, c=2, t;  
while(a<b<c) { t=a; a=b; b=t; c--; }  
printf("%d, %d, %d", a, b, c);  
A) 1, 2, 0 B) 2, 1, 0 C) 1, 2, 1 D) 2, 1, 1 A
- 46 设a, b和c都是int型变量, 且a=3, b=4, c=5, 则下列表达式中, 值为0的表达式是\_\_\_\_\_  
A) 'a'&&'b' B) a<=b C) a||b+c&&b-c D) !((a<b)&&!c||1) D
- 47 设x, y, z和k都是int型变量, 则执行表达式:x=(y=4, z=16, k=32)后, x的值为 A) 4 B) 16 C) 32 D) 52 C
- 48 设x=3, y=-4, z=6, 写出表达式!(x>y)+(y!=z)|| (x+y)&&(y-z)的结果 A) 0 B) 1 C) -1 D) 6 B
- 49 设x和y均为int型变量, 则以下语句:x+=y; y=x-y; y=x-y; x-=y;的功能是\_\_\_\_\_  
A、把x和y按从大到小排列 B、把x和y按从小到大排列 C、无确定结果 D、交换x和y中的值 D

- 50 设有 `int x=11;` 则表达式 `(x++ * 1/3)` 的值是 A) 3 B) 4 C) 11 D) 12 A
- 51 设有如下的变量定义: 则以下符合C语言语法的表达式是  
`int i =8, k , a, b ;`  
`unsigned long w=5;`  
`double x=1, 42, y=5. 2;`  
A、 `a+=a-=(b=4)*(a=3)` B、 `x%(-3);` C、 `a=a*3=2` D、 `y=float(i)` A
- 52 设有如下定义: `char *aa[2]={ "abcd", "ABCD" }`; 则以下就法中正确的是  
A) aa数组成元素的值分别是"abcd"和"ABCD"  
B) aa是指针变量, 它指向含有两个数组元素的字符型-维数组  
C) aa数组的两个元素分别存放的是含有4个字符的-维字符数组的首地址  
D) aa数组的两个元素中各自存放了字符'a'和'A'的地址 D
- 53 设有如下定义: `int x=10, y=3, z;` 则语句`printf("%d\n", z=(x*y, x/y));` 的输出结果是  
A) 1 B) 0 C) 4 D) 3 D
- 54 设有数组定义: `char array[ ]="China";` 则数组 array所占的空间为  
A) 4个字节 B) 5个字节 C) 6个字节 D) 7个字节 C
- 55 设有说明语句:`char a=' \72';`则变量a  
A) 包含1个字符 B) 包含2个字符 C) 包含3个字符 D) 说明不合法 A
- 56 语句`while(!E)`中的表达式!E等价于:  
A) `E==0` B) `E!=1` C) `E!=0` D) `E==1` A
- 57 下列if语句中, 不正确的是\_\_\_\_\_。  
A、 `if(x>y);` B、 `if(x==y) x+=y;` C、 `if(x!=y) scanf("%d",&x) else x=1;` D、 `if(x<y) {x++;y++}` C
- 58 面程序段的运行结果是  
`if (n=0);`  
`while(n++<=2); printf("%d",n);`  
A) 2 B) 3 C) 4 D)有语法错 B
- 59 下列标识符中不合法的是\_\_\_\_\_ A、 `s_name` B、 `_e` C、 `fox` D、 `3DS` D
- 60 下列不正确的转义字符是 A) `'\\'` B) `'\"'` C) `'074'` D) `'\0'` C
- 61 下面程序的输出结果是\_\_\_\_\_ A、 2 B、 3 C、 4 D、 1  
`main()`  
`{ int a=1, b=2;`  
`printf("%d\n", a=a+1, a+6, b+2); }` A
- 62 下面程序的输出结果是\_\_\_\_\_  
`int i=0, j=0, a=6;`  
`if((++i>0) || (++j>0)) a++;`  
`printf("i=%d, j=%d, a=%d\n", i, j, a);`  
A、 `i=0, j=0, a=6` B、 `i=1, j=0, a=7` C、 `i=1, j=1, a=6` D、 `i=1, j=1, a=7` B
- 63 下列程序段的输出结果为\_\_\_\_\_ A、 `y=-1` B、 `y=0` C、 死循环 D、 `y=9`  
`int y=1;`  
`while(y--);`  
`printf("y=%d\n", y);` A
- 64 下列程序执行后的输出结果是 A) 3 B) 6 C) 9 D) 随机数  
`main()`  
`{ int a[3][3], *p, i;`  
`p=&a[0][0];`  
`for(i=0; i<9; i++) p[i]=i+1;`  
`printf("%d \n", a[1][2]);`  
`}` B
- 65 下列程序执行后的输出结果是 A) G B) H C) i D) J  
`main()`  
`{ int x='f'; printf("%c \n", 'A'+(x-'a'+1)); }` A
- 66 下列程序执行后的输出结果是(小数点后只写一位)  
`main()`  
`{ double d; float f; long l; int i;`  
`i=f=l=d=20/3;`  
`printf("%d %ld %f %f \n", i, l, f, d);`  
`}`  
A) 6 6 6.0 6.0 B) 6 6 6.7 6.7 C) 6 6 6.0 6.7 D) 6 6 6.7 6.0 A
- 67 下列可作为C语言赋值语句的是 A) `x=3, y=5` B) `a=b=6` C) `i--;` D) `y=int(x);` C
- 68 下列描述中不正确的是  
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  
C) 可以对整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 C
- 69 下列说法中正确的是\_\_\_\_\_。  
A、 在switch语句中一定要使用break语句 B、 在switch语句中不一定要使用break语句  
C、 break语句是switch语句的一部分 D、 break只能用于switch语句中 B
- 70 下列运算符其优先级最高的是\_\_\_\_\_。 A、 `||` B、 `&&` C、 `+` D、 `=` C

- 71 下面的DO循环中，一共循环( )次。 (A) 1 (B) 6 (C) 4 (D) 5
- ```

M=5
N=1
do
{N=N+1;
}while(N<=M);

```
- D
- 72 #include <math.h>  
main()  
{  
int a=1,b=4,c=2;  
float x=5.5,y=9.0,z;  
z=(a+b)/c+sqrt((double)y)\*1.2/c+x;  
printf("%f\n",z);  
}
- C
- 则上面程序的输出结果是\_\_\_\_\_ A、数据溢出 B、9.3 C、9.300000 D、0.93
- 73 下面程序的输出结果是\_\_\_\_\_
- ```

int a,b,c;
a=(b=(c=10)+5)-5;
printf("a,b,c=%d,%d,%d",a,b,c);
c=a+0; b=(a+10);
printf("a,b,c=%d,%d,%d",a,b,c);

```
- C
- A、a,b,c=0,10,10 a,b,c=10,15,10 B、a,b,c=10,15,10 a,b,c=10,15,10  
C、a,b,c=10,15,10 a,b,c=0,10,0 D、a,b,c=10,15,10 a,b,c=10,15,15
- 74 写出下列语句执行后a,b的值。
- ```

int a=6,b;
b=(++a)+(a++);

```
- A
- A、a=8, b=14 B、a=7, b=8 C、a=14, b=14 D、a=8, b=8
- 75 请读程序： 则下面程序的输出结果是\_\_\_\_\_
- ```

m=1234.123;
printf("%-8.3f\n",m);
printf("%10.3f\n",m);

```
- B
- A、1234.12300 B、1234.123 C、1234.123 D、1234.12300  
1234.123 1234.123 1234.12300 1234.123
- 76 一个C语言的语句至少应包一个\_\_\_\_\_ A、{} B、逗号 C、分号 D、什么都不要
- C
- 77 一个完整的C源程序是\_\_\_\_\_。
- A、要由一个主函数(或)一个以上的非主函数构成  
B、由一个且仅由一个主函数和零个以上(含零)的非主函数构成  
C、要由一个主函数和一个以上的非主函数构成  
D、由一个且只有一个主函数或多个非主函数构成
- B
- 78 下面程序的运行结果是
- ```

#include <stdio.h>
main()
{char a[]="morming",t;
int i,j=0;
for(i=1;i<7;i++) if(a[j]<a[i]) j=i;
t=a[j];a[j]=a[7];
a[7]=a[j];puts(a);
}


```
- B
- A) mogninr B) mo C) morning D) mornin
- 79 以下for语句构成的循环执行了( )次。 (A) 5 (B) 6 (C) 3 (D) 9
- ```

#include<stdio.h>
#define N 1
#define M N+1
#define NUM (M+1)*M/2
main( )
{int i,n=0;
for (i=1;i<=NUM;i++) n++;
printf("%d",n);
}

```
- C
- 80 以下程序的输出结果是\_\_\_\_\_ (A) 不确定的 (B) 0 (C) -1 (D) 1
- ```

#include<stdio.h>
main( )
{
printf("%d\n",NULL);
}

```
- B

- 81 在调用函数时，以下描述中不正确的是\_\_\_\_\_。  
A、调用函数时，实参可以是表达式 C、调用函数时，实参与形参的原型必须一致 D  
B、调用函数时，将为形参分配内存单元 D、调用函数时，实参与形参可以共用内存单元
- 82 有以下程序段，则while 循环执行的次数是  
int k=0;  
while(k=1) k++;  
A) 无限次 B) 有语法错，不能执行 C) 一次也不执行 D) 执行1次 A
- 83 有以下函数，该函数的返回值是  
char fun(char \*p)  
{ return p; }  
A. 无确切的值 B. 形参p中存放的地址值 B. 一个临时存储单元的地址 D. 形参p自身的地址值 B
- 84 若有说明:int a[ ][3]={1,2,3,4,5,6,7};则a数组第1维的大小是\_\_\_\_。  
A) 2 B) 3 C) 4 D) 无确定值 B
- 85 语句:printf("%d", (a=2)&&(b=-2));的输出结果是\_\_\_\_  
A) 无输出 B) 结果不确定 C) -1 D) 1 D
- 86 运算完下面的C语言程序段以后，a的值是\_\_\_\_，b的值是\_\_\_\_，c的值是\_\_\_\_\_  
int x=10,y=9;  
int a,b,c;  
a=(--x==y++)?--x:++y;  
b=x++;  
c=y;  
A、6 9 13 B、8 7 11 C、8 8 10 D、8 7 10 C
- 87 在C语言程序中，注释部分从哪开始，到哪结束？  
A、从(到)结束 B、从REM开始无结束标志 D  
C、无起始标志 D、从/\*开始，\*/结束
- 88 在C语言提供的合法的关键字是 A) swicth B) cher C) Case D) default D
- 89 在C语言中，不正确的int类型的常数是\_\_\_\_ A) 32768 B) 0 C) 037 D) 0xAF A
- 90 在C语言程序中，当while语句构成的循环中的条件为\_\_\_\_时，结束循环。  
A、0 B、1 C、true D、非0 A
- 91 在C语言中，函数的隐含存储类别是 A) auto B) static C) extern D) 无存储类别 C
- 92 在C语言中，合法的长整型常数是 A) 0L B) 4962710 C) 324562& D) 216D A
- 93 在C语言中，合法的字符常量是 A) '\084' B) '\x43' C) 'ab' D) "\0" B
- 94 在C语言中，若对函数类型未加明显式说明，则函数的隐含类型是\_\_\_\_类型  
A) void B) double C) int D) char C
- 95 在C语言中，形参的缺省存储类是 A) auto B) register C) static D) extern  A
- 96 在C语言中，要求运算数必须是整型的运算符是 A) % B) / C) < D) ! A
- 97 在宏定义 #define PI 3.14159中，用宏名PI代替一个  
A) 单精度数 B) 双精度数 C) 常量 D) 字符串 C
- 98 在C语言程序中，可以用做变量名的是\_\_\_\_ A) 1e5 B) a1 C) int D) \*p B
- 99 在以下一组运算符中，优先级最高的运算符是 A) <= B) = C) % D) && C
- 100 字符(char)型数据在微机内存中的存储形式是 A. 反码 B. 补码 C. EBCDIC码 D. ASCII码 D
- 101 执行语句: for (i=1;i++<4;);后,变量i的值是 A 3 B 4 C 5 D不定 C
- 102 指出下面正确的输入语句是( )  
A:scanf("a=b=%d",&a,&b) B:scanf("a=%d,b=%f",&a,&b)  
C:scanf("%3c",c) D:scanf("%5.2f",&a) B
- 103 执行下面的程序后，a的值为 A) 7 B) 8 C) 9 D) 10  
main()  
{ int a,b;  
for(a=1,b=1;a<=100;a++)  
{ if(b>=20) break;  
if(b%3==1) { b+=3; continue; }  
b-=5; } }
- 104 在执行以下程序时，如果从键盘上输入：ABCdef〈回车〉，则输出为  
A)ABCdef B)abcdef C)abc D)DEF  
main()  
{ chsr ch;  
while((ch=getchar())!='\n')  
{ if(ch>='A'&& ch<='Z') {ch=ch+32;printf("%c",c);} } }
- 105 在c语言中，引用数组元素时，其数组下标的数据类型允许是\_\_\_\_。  
A) 整型常量 B) 整型表达式 C  
c) 整型常量或整型表达式D) 任何类型的表达式
- 106 在以下定义中，正确的说明语句为\_\_\_\_  
A) int a['a'] B) int a[3,4] C) int a[ ][ ] D) int \*a[10] D

- 107 执行下列程序时输入:123<空格>456<空格>789<回车>, 输出结果是  
 A) 123,456,789    B) 1,456,789    C) 1,23,456,789    D) 1,23,456  

```
main()
{ char s[100]; int c, i;
scanf("%c",&c); scanf("%d",&i); scanf("%s",s);
printf("%c,%d,%s\n",c,i,s);
}
```

D
- 108 执行下面程序片段的的结果是 A. 打印出321 B. 打印出23 C. 不打印任何内容 D. 陷入死循环  

```
int x=23;
do
{ printf("%2d",x--);}while(!x);
```

B
- 109 在说明语句: `int *f();` 中, 标识符 `f` 代表的是\_\_\_\_  
 A) 一个用于指向整型数据的指针变量    B) 一个用于指向一维数组的行指针  
 C) 一个用于指向函数的指针变量    D) 一个返回值为指针型的函数名  
D
- 110 在下列选项中, 不正确的赋值语句是\_\_\_\_  
 A) `++t;`    B) `n1=(n2=(n3=0));`    C) `k=i==j;`    D) `a=b+c=1;`  
D
- 111 在C语言中, 若需一变量只在本文件中所有函数使用, 则该变量的存储类别是\_\_\_\_\_  
 A、`extern`    B、`register`    C、`auto`    D、`static`  
D
- 112 在C语言程序中, 当do-while语句构成的循环中的条件为\_\_\_\_时, 结束循环。  
 A、0    B、1    C、true    D、非0  
A
- 113 在16位C编译系统上, 若定义 `long a;`, 则能给 `a` 赋40000的正确语句是  
 A) `a=20000+20000;`    B) `a=4000*10;`    C) `a=30000+10000;`    D) `a=4000L*10L;`  
D
- 114 阅读以下程序:  

```
main()
{ int x;
scanf("%d",&x);
if(x--<5) printf("%d",x);
else printf("%d",x++);
}
```

B  
 程序运行后, 如果从键盘上输入5, 则输出结果是    A) 3    B) 4    C) 5    D) 6
- 115 运行以下程序后, 如果从键盘上输入 `china#`<回车>, 则输出结果为  
 A、2,0    B、5,0    C、5,5    D、2,5  

```
main( )
{ int v1=0,v2=0;
char ch ;
while ((ch=getchar())!='#')
switch (ch )
{ case 'a':
case 'h':
default: v1++;
case '0':v2++;
}
printf("%d,%d\n",v1,v2);
```

C
- 116 请读程序:  

```
main()
{
int a=1,b=2,c=3;
++a;
c+=++b;
{
int b=4,c;
c=b*3;
a+=c;
printf("first:%d,%d,%d\n",a,b,c);
a+=c;
printf("Second:%d,%d,%d\n",a,b,c);
}
printf("third:%d,%d,%d\n",a,b,c);
}
```

C  
 该程序的输出是\_\_\_\_\_。  
 A、first: 12,4,12    Second: 26,4,12    third: 4,3,6  
 B、first: 14,4,12    Second: 26,12,12    third: 26,3,4  
 C、first: 14,4,12    Second: 26,4,12    third: 26,3,6  
 D、first: 14,4,12    Second: 26,6,12    third: 26,4,6



- 117 阅读程序：  
#include<stdio.h>  
main()  
{ int a[10]={1,2,3,4,5,6,7,8,9,0},\*p;  
p=a;  
printf("%x\n",p);  
printf("%x\n",p+9); }  
该程序有两个printf 语句，如果第一个printf 语句输出的是194，则第二个printf语句的输出结果是  
A) 203      B) 204      C) 1a4      D) 1a6
- 118 与  $y=(x>0?1:x<0?-1:0)$  ;的功能相同的if语句是  
A) if (x>0) y=1;      B) if(x)      C) y=-1      D) y=0;  
else if(x<0) y=-1;      if(x>0) y=1;      if(x)      if(x>=0)  
else y=0;      else if(x<0) y=-1;      if(x>0) y=1;      if(x>0) y=1;  
else y=0;      else if(x==0) y=0;      else y=-1;  
else y=-1;
- 119 以下合法的赋值语句是      A) x=y=100      B) d--;      C) x+y;      D) c=int(a+b);
- 120 以下函数返回a所指数组中最小的值所在的下标值  
fun(int \*a, int n)  
{ int i,j=0,p;  
p=j;  
for(i=j;i<n;i++)  
if(a[i]<a[p]) \_\_\_\_\_;  
return(p);  
}  
在下划线处应填入的是      A) i=p      B) a[p]=a[i]      C) p=j      D) p=i
- 121 以下函数值的类型是      A) int      B) 不确定      C) void      D) float  
fun ( float x )  
{ float y;  
y= 3\*x-4;  
return y;  
}
- 122 以下说法中正确的是  
A) C语言程序总是从第一个的函数开始执行  
B) 在C语言程序中，要调用的函数必须在main()函数中定义  
C) C语言程序总是从main()函数开始执行  
D) C语言程序中的main()函数必须放在程序的开始部分
- 123 以下所列的各函数首部中，正确的是  
A) void play(var a:integer,var b:integer)      B) void play(int a,b)  
C) void play(int a,int b)      D) sub play(a as integer,b as integer)
- 124 以下叙述正确的是  
A) 输入项可以是一个实型常量，如： scanf("%f".3.5);  
B) 只有格式控制，没有输入项，也能正确输入数据到内存，例如： scanf("a=%d,b=%d");  
C) 当输入一个实型数据时，格式控制部分可以规定小数点后的位数，例如： scanf("%4.2f",&f);  
D) 当输入数据时，必须指明变量地址，例如： scanf("%f",&f);
- 125 以下叙述正确的是  
A) do-while语句构成的循环不能用其它语句构成的循环来代替。  
B) do-while语句构成的循环只能用break语句退出。  
C) 用do-while语句构成的循环，在while后的表达式为非零时结束循环。  
D) 用do-while语句构成的循环，在while后的表达式为零时结束循环。
- 126 以下叙述中不正确的是  
A) 在C中，函数中的自动变量可以赋初值，每调用一次，赋一次初值。  
B) 在C中，在调用函数时，实在参数和对应形参在类型上只需赋值兼容。  
C) 在C中，外部变量的隐含类别是自动存储类别。  
D) 在C中，函数形参可以说明为register变量。
- 127 以下叙述中不正确的是\_\_\_\_\_  
A) 在不同的函数中可以使用相同名字的变量      B) 函数中的形式参数是局部变量  
C) 在一个函数内定义的变量只在本函数范围内有效  
D) 在一个函数内的复合语句中定义的变量在本函数范围内有效

128 以下程序运行后,输出结果是      A) 1000      B) 1001      C) 00110      D) 10100

```
main ()
{ int y=18,i=0,j,a[8];
do
{ a[i]=y%2;i++;
y=y/2;
} while(y>=1);
for(j=i-1;j>0;j--)printf("%d",a[j]);
printf("\n");
}
```

B

129 以下程序运行后,如果从键盘上输入:

book<回车>

book<空格><回车>

则输出结果是      A) \*8      B) #9      C) #6      D) \*9

```
#include
main()
{char a1[80],a2[80],*s1=a1,*s2=a2;
gets(s1); gets(s2);
if(!strcmp(s1,s2)) printf("*");
else printf("#");
printf("%d\n",strlen(strcat(s1,s2)));
}
```

B

130 以下程序运行后,输出结果是      A) 6385      B) 22      C) 33      D) 693825

```
main()
{ char ch[2][5]={"693","825"},*p[2];
int i,j,s=0;
for (i=0;i<2;i++) p[i]=ch[i];
for (i=0;i<2;i++)
for (j=0;p[i][j]>='0' && p[i][j]<='9';j+=2)
s=10*s+p[i][j]-'0';
printf("%d\n",s);
}
```

A

131 以下程序运行后,输出结果是      A) 84      B) 99      C) 95      D) 44

```
int d=1;
fun (int p)
{ int d=5;
d+=p++;
printf("%d",d);
}
main()
{
int a=3;
fun(a);
d+=a++;
printf("%d\n",d);
}
```

A

132 以下程序执行后a的值是      A) 4      B) 1      C) 0      D) 运行时出错, a无定值

```
main()
{ int a,k=4,m=6,*p1=&k,*p2=&m;
a=p1==&m;
printf("%d\n",a);
}
```

C

133 以下程序执行后sum的值是  
A) 15 B) 14 C) 不确定 D) 0

```
min()
{ int i, sum;
for(i=1;i<6;i++) sum+=i;
printf("%d\n",sum);
}
```

C



- 134 以下程序中,while循环的循环次数是 A. 1 B. 10 C. 6 D. 死循环, 不能确定次数  
 c  

```
main()
{ int i=0;
while(i<10)
{ if(i<1) continue;
if(i==5) break;
i++;
}
.....
}
```

 D
- 135 以下定义语句中, 错误的是  
 A) int a[]={1,2}; B) char \*a[3]; C) char s[10]="test"; D) int n=5,a[n]; D
- 136 以下对C语言函数的有关描述中, 正确的是  
 A) 在C中, 调用函数时, 只能把实参的值传送给形参, 形参的值不能传送给实参  
 B) C函数既可以嵌套定义又可以递归调用 C) 函数必须有返回值, 否则不能使用函数  
 D) C程序中有调用关系的所有函数必须放在同一个源程序文件中 A
- 137 以下对枚举类型名的定义中正确的是\_\_\_\_  
 A) enum a={one,two,three}; B) enum a {one=9,two=-1,three};  
 C) enum a={"one","two","three"}; D) enum a {"one","two","three"}; B
- 138 以下各选项企图说明一种新的类型名, 其中正确的是  
 A. typedef v1 int; B. typedef v2=int; C. typedef v1 int v3; D. typedef v4: int; C
- 139 以下选项中, 不能正确赋值的是  
 A) char s1[10];s1="Ctest"; B) char s2[]={ 'C', 't', 'e', 's', 't' };  
 C) char s3[20]="Ctest"; D) char \*s4="Ctest\n" A
- 140 以下选项中, 非法的字符常量是 A) '\t' B) '\17' C) "n" D) '\xaa' C
- 141 以下选项中不属于C语言的类型的是  
 A) signed short int B) unsigned long int  
 C) unsigned int D) long short D
- 142 以下选项中合法的字符常量是 A) "B" B) '\010' C) 68 D) D B
- 143 以下循环体的执行次数是 A) 3 B) 2 C) 1 D) 0
- main()  
 { int i,j;  
 for(i=0,j=1; i<=j+1; i+=2, j--)printf("%d \n",i);  
 } C
- 144 以下有4组用户标识符, 其中合法的一组是 A) for B) 4d C) f2\_G3 D) if C
- 145 用十进制数表示表达式: 12/012的运算结果是 A) 1 B) 0 C) 14 D) 12 A
- 146 设有 int a = 1,b=2,c =3, d =4, m=2, n=2;  
 执行(m= a>b)&&(n=c>d)后n的值为:  
 A) 1 B) 2 C) 3 D) 4 B
- 147 以下程序该程序的输出的结果是 A) 12 B) 13 C) 14 D) 15  

```
int runc(int a,int b)
{ return(a+b);}
main0
{ int x=2,y=5,z=8,r;
r=func(func(x,y),z);
printf("%d\n",r);
}
```

 D
- 148 以下程序的输出结果是 A) 2 B) 4 C) 6 D) 8  

```
long fib(int n)
{ if(n>2) return(fib(n-1)+fib(n-2));
else return(2);
}
main0
{ printf("%d\n",fib(3));
}
```

 B
- 149 main()  
 { char s[]="ABCD", \*p;  
 for(p=s+1; p<s+4; p++) printf ("%s\n",p);  
 }  
 以上程序的输出结果是 A) ABCD B) A C) B D) BCD  
 BCD B C CD  
 CD C D D  
 D D D D D

- 150 以下程序的输出结果是      A) 14      B) 19      C) 20      D) 21
- ```
main()
{inta[3][3]={ {1,2}, {3,4}, {5,6}}, i, j, s=0;
for(i=1;i<3;i++)
for(j=0;j<i;j++)s+=a[i][j]
printf("%d\n",s);
}
```
- 151 以下程序的输出结果是      A) 0      B) 1      C) 2      D) 3
- ```
main0
{ int a=2,b=-1,c=2;
if(a<b)
if(b<0) c=0;
else c++
printf("%d\n",c);
}
```
- 152 以下程序的输出结果是      A) 6      B) 3      C) 死循环      D) 0
- ```
main0
{ int i,sum;
for(i=1;i<=3;sum++) sum+=i;
printf("%d\n",sum);
}
```
- 153 以下程序的输出结果是      A) 不确定的值      B) 2      C) 1      D) 0
- ```
main0
{ int n[5]={0,0,0}, i, k=2;
for(i=0;i<k;i++) n[i]=n[i]+1;
printf("%d\n",n[k]);
}
```
- 154 以下程序的输出结果是      A) 987      B) 876      C) 8765      D) 9876
- ```
main0
{ int n=9;
while(n>6) {n--;printf("%d",n);}
}
```
- 155 以下程序的输出结果是      A) a=2, b=1      B) a=1, b=1      C) a=1, b=0      D) a=2, b=2
- ```
main0
{ int x=1, a=0, b=0;
switch(x) {
case 0: b++;
case 1: a++;
case 2: a++;b++;
}
printf("a=%d,b=%d\n", a, b);
}
```
- 156 以下程序运行该程序的输出结果是      A) 3 4      B) 4 2      C) 4 3      D) 3 3
- ```
main0
{ int y=3, x=3, z=1;
printf("%d %d\n", (++x, y++), z+2);
}
```
- 157 以下程序段执行后, a的值为      A) 12      B) 11      C) 10      D) 编译出错
- ```
int *p, a=10, b=1
p=&a; a=*p+b;
```
- 158 有如下说明, 则数值为9的表达式是      A) \*p+9      B) \*(p+8)      C) \*p+=9      D) p+8
- ```
int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=a;
```
- 159 设有以下宏定义, 则执行语句: z=2\*(N+Y(5+1)); 后, z的值为A) 出错 B) 42 C) 48 D) 54
- ```
#define N 3
#define Y(n) ( (N+1)*n)
```
- 160 以下程序的输出结果是      A) 258 B) 741 C) 852 D) 369
- ```
main()
{ int i, a[10];
for(i=9;i>=0;i--) a[i]=10-i;
printf("%d%d%d", a[2], a[5], a[8]);
}
```

- 161 `main()`  
`{ int i,k,a[10],p[3];`  
`k=5;`  
`for (i=0;i<10;i++) a[i]=i;`  
`for (i=0;i<3;i++) p[i]=a[i*(i+1)];`  
`for (i=0;i<3;i++) k+=p[i]*2;`  
`printf("%d\n",k);`  
`}`  
 以上程序的输出结果是      A) 20      B) 21      C) 22      D) 23      B
- 162 `main()`  
`{ int num= 0;`  
`while(num<=2)`  
`{ num++; printf("%d\n",num);`  
`}`  
 以上程序的输出结果是      A) 1      B) 1      C) 1      D) 1      B  
    2            2            2  
    3            3  
    4
- 163 在C语言中,1维数组的定义方式为:      类型说明符      数组名      \_\_\_\_\_ 。      A  
 A) [常量表达式]      B) [整形表达式]  
 c) [ 整型常量]或[整型表达式]      D) [整型常量]
- 164 以下程序的输出结果是      A) 100200      B) 57      C) 200100      D) 75
- ```
int a, b;
void fun()
{ a=100; b=200; }
main()
{ int a=5, b=7;
  fun();
  printf("%d%d \n", a,b);
}
```
- 165 以下程序的输出结果是      D  
`main( )`  
`{ int a=-1,b=4,k;`  
`k=(++a<0)&&!(b--<=0);`  
`printf("%d%d%d\n",k,a,b); }`  
 A) 104 B) 103 C) 003 D) 004
- 166 以下程序的输出结果是      A) a B) Z C) z-25 D) y      A  
`main()`  
`{ char c='z';`  
`printf("%c",c-25);`  
`}`
- 167 以下程序的输出结果是      A) 9 9 B) 5 20 C) 13 20 D) 20 20      B  
`main()`  
`{ char st[20]= "hello\0\t\\\"";`  
`printf("%d %d \n",strlen(st),sizeof(st));`  
`}`
- 168 以下程序的输出结果是      A  
`main()`  
`{ int a[4][4]={ {1,3,5}, {2,4,6}, {3,5,7} };`  
`printf("%d%d%d%d\n",a[0][3],a[1][2],a[2][1],a[3][0];`  
`}`  
 A) 0650 B) 1470 C) 5430 D) 输出值不定
- 169 以下程序段的输出结果是      A) 9      B) 12      C) 13      D) 14      A  
`char s[]="\\141\\141abc\t";`  
`printf ("%d\n",strlen(s));`
- 170 以下程序段的输出结果是      A) 12      B) 34      C) 1234      D) 提示出错、无结果      C  
`int a=1234;`  
`printf("%2d\n",a);`
- 171 以下程序段的输出结果是      A) 1      B) 3 0      C) 1      -2      D) 死循环      C  
`int x=3;`  
`do { printf("%3d",x-=2);} while(!(--x));`

172 以下程序段的执行结果是

- A) a=12 y=12      B) a=12 y=12      C) a=12<sup>c</sup>y=12      D) a=12 y=12  
a=14 y=16      a=16 y=28      a=14 y=26  
a=16 y=20      a=14 y=44  
a=18 y=24

```
int a, y;  
a=10; y=0;  
do  
{ a+=2; y+=a;  
printf("a=%d y=%d\n", a, y);  
if(y>20) break;  
{ while(a=14);
```

B

173 以下程序段给数组所有的元素输入数据，请选择正确答案填入。

```
#include<stdio.h>  
main()  
{ int a[10], i=0;  
while(i<10) scanf("%d", _____ );  
:  
}  
A) a+(i++)    B) &a[i+1]  
C) a+i        D) &a[++i]
```

A

174 以下程序输出结果是

A、7      B、6      C、5      D、4  
main ( )  
{ int m=5;  
if (m++>5) print ("%d\n", m);  
else printf ("%d\n", m--);  
}

B

175 以下程序运行后, 如果从键盘上输入ABCDE<回车>, 则输出结果为

A、8      B、7  
C、6      D、5  
fuch (char str [ ] )  
{ int num =0;  
while (\*(str +num )!='\0') num ++;  
return(num );  
}  
main()  
{ char str [10], \*p=str ;  
gets(p); printf("%d\n", func(p));  
}

D

176 以下程序运行后, 输出结果是

A 49.5    B 9.5    C 22.0    D 45.0  
#include  
#define PT 5.5  
#define s(x) pt\*x\*x  
main()  
{ int a=1, b=2;  
print ("%4.1f\n", S(a+b));  
}

B

177 以下程序运行后, 输出结果是

A 8, 15    B 8, 16    C 8, 17    D 8, 8  
func (int a, int b)  
{int m=0 , i=2;  
i+=m+1;  
m=i+a+b ;  
return(m);  
}  
main()  
{  
int k=4, m=1, p;  
p=func (k, m); printf("%d, ", p);  
p=func (k, m); printf("%d\n", p);  
}

D

- 178 以下程序运行后,输出结果是    A、ABCDEFGH    B、CDG    C、abcdefgh    D、abCDefGh
- ```
main()
{ static char a[]="ABCDEFGH",b[]="abCDefGh";
char *p1,*p2;
int k;
p1=a; p2=b;
for(k=0;k<=7;k++)
if (*(p1+k)==*(p2+k))
printf("%c",*(p1+k));
printf("\n");
}
```
- 179 以下程序运行后,输出结果是    A. cde    B. 字符c的ASCLL码值    C. 字符C的地址    D. 出错
- ```
main( )
{ char *s="abcde";
s+=2;
print("%ld \n",s);
}
```
- 180 以下程序的输出结果是    A) 3    B) 4    C) 1    D) 9
- ```
main()
{ int b[3][3]={0,1,2,0,1,2,0,1,2},i,j,t=1;
for(i=0;i<3;i++)
for(j=i;j<=i;j++) t=t+b[i][b[j][j]];
printf("%d\n",t);
}
```
- 181 以下程序的输出结果是    A) 19 B) 17 C) 15 D) 12
- ```
#define M(x,y,z) x*y+z
main()
{ int a=1,b=2, c=3;
printf("%d\n", M(a+b,b+c, c+a));
}
```
- 182 以下程序的输出结果是    A) abcdeg B) bcde C) ABCDE D) BCDEFG
- ```
#include"ctype.h"
space (char *str)
{ int i,t;char ts[81];
for(i=0,t=0;str[i]!='\0';i+=2)
if(! isspace(*str+i)&&(*str+i)!='a'))
ts[t++]=toupper(str[i]);
ts[t]='\0';
strcpy(str,ts); }
main( )
{ char s[81]="a b c d e f g";
space(s);
puts(s); }
```
- 183 以下程序的输出结果是    A) ##### B) ##### C) \*\*\*\*\* D) \*\*\*\*\*
- ```
main( )
{ int i;
for(i=1;i<6;i++)
{ if(i%2) { printf("#");continue; }
printf("*");}
printf("\n");
}
```
- 184 以下程序的输出结果是    A) 0    B) 1    C) 2    D) 3
- ```
main()
{ int x=0.5; char z='a';
printf("%d\n", (x&&1)&&(z<'z')) ); }
```
- 185 以下程序的输出结果是    A) -1 1    B) 0 1    C) 1 0    D) 0 0
- ```
main( )
{ int a= -1,b=1,k;
if( (++a<0) && !(b--<=0))
printf("%d %d\n",a,b);
else printf("%d %d\n",b,a);
}
```

- 186 以下程序的输出结果是 A) 1 B) 2 C) 3 D) 4
- ```

long fun( int n)
{ long s;
  if(n= =1 || n= =2) s=2;
  else s=n-fun(n-1);
  return s;
}

main()
{ printf("%ld\n", fun(3)); }

```
- 187 若有以下调用语句,则不正确的fun函数的首部是
- A) void fun(int m, int x[]) B) void fun(int s, int h[41])  
 C) void fun(int p, int \*s) D) void fun(int n, int a)
- ```

main()
{ ...
  int a[50],n;
  ...
  fun(n, &a[9]);
  ...
}

```
- 188 以下程序的输出结果是
- A) 10 1 9 2 B) 9 8 7 6 C) 10 9 9 0 D) 10 10 9 1
- ```

main( )
{ int x=10,y=10,i;
  for(i=0;x>8;y=++i)
  printf("%d,%d ",x--,y);
}

```
- 189 以下程序的输出结果是 A) 123456780 B) 123 456 780 C) 12345678 D) 147
- ```

main( )
{ char ch[3][4]={"123","456","78"}, *p[3]; int i;
  for(i=0;i<3;i++) p[i]=ch[i];
  for(i=0;i<3;i++) printf("%s",p[i]);
}

```
- 190 以下程序的输出结果是 A) 16 B) 2 C) 9 D) 1
- ```

#define SQR(X) X*X
main()
{ int a=16, k=2, m=1;
  a/=SQR(k+m)/SQR(k+m);
  printf("%d\n",a);
}

```
- 191 以下程序的输出结果是
- A) 17,021,0x11 B) 17,17,17 C) 17,0x11,021 D) 17,21,11
- ```

main( )
{ int k=17;
  printf("%d,%o,%x \n",k,k,k);
}

```
- 192 假设所有变量均为整型,则表达式(a=2,b=5,b++,a+b)的值是:
- A) 7 B) 8 C) 6 D) 2
- 193 以下程序的输出结果是 A) 2 0 B) 3 1 C) 3 2 1 D) 2 1 0
- ```

main( )
{ int n=4;
  while(n-->0) printf("%d ",n);
}

```
- 194 以下程序的输出结果是 A) 20 B) 21 C) 22 D) 23
- ```

main()
{ int i, k, a[10], p[3];
  k=6;
  for (i=0;i<10;i++) a[i]=i;
  for (i=0;i<3;i++) p[i]=a[i *(i+1)];
  for (i=0;i<3;i++) k+=p[i] *2;
  printf("%d\n",k);
}

```

- 195 以下程序的输出结果是    A) 6 9 9    B) 6 6 9    C) 6 15 15    D) 6 6 15
- ```

int d=1;
fun(int p)
{ static int d=5;
d+=p;
printf("%d ",d);
return(d);
}
main( )
{ int a=3; printf("%d \n",fun(a+fun(d))); }

```
- 196 以下程序的输出结果是    A) 9    B) 6    C) 36    D) 18
- ```

#define f(x) x*x
main( )
{ int a=6,b=2,c;
c=f(a)/f(b);
printf("%d \n",c);
}

```
- 197 以下程序的输出结果是
- A) ABCD B) ABCD C) EFG D) EFGH
- FGH      EFG      JK      IJKL
- KL      IJ      0
- M
- ```

main()
{ char w[][10]={ "ABCD","EFGH","IJKL","MNOP"},k;
for(k=1;k<3;k++) printf("%s\n",w[k]);
}

```
- 198 以下程序的输出结果是    A) 编译不通过,无输出 B) aceg    C) acegi    D) abcdefghi
- ```

main( )
{ int i;
for(i='A';i<'I';i++,i++) printf("%c",i+32);
printf(" \n");
}

```
- 199 以下程序的输出结果是    A) 4 2 1 1    B) 0 0 0 8    C) 4 6 7 8    D) 8 8 8 8
- ```

main( )
{ char *s="12134211"; int v[4]={0,0,0,0},k,i;
for(k=0;s[k];k++)
{ switch(s[k])
{ case '1': i=0;
case '2': i=1;
case '3': i=2;
case '4': i=3;
}
v[i]++;
}
for(k=0;k<4;k++) printf("%d ",v[k]);
}

```
- 200 以下程序的输出结果是    A) 6    B) 67890    C) '6'    D) 789
- ```

main( )
{ char a[10]={ '1','2','3','4','5','6','7','8','9','0'},*p;
int i;
i=8;
p=a+i;
printf("%s\n",p-3);
}

```
- 201 以下程序的输出的结果是    A) 3 3    B) 2 2    C) 2 6    D) 2 5
- ```

int x=3;
main()
{ int i;
for (i=1;i<x;i++) incre();
}

ncre()
{ static int x=1;
x*=x+1;
printf(" %d",x);
}

```



- 202 以下程序的输出结果是      A) 21      B) 32      C) 33      D) 11
- ```
#include <stdio.h>
main()
{ int i=0,a=0;
while(i<20)
{ for(;;)
{ if((i%10)==0) break;
else i--;
}
i+=11; a+=i;
}
printf("%d\n",a);
}
```
- 203 以下程序的输出结果是      A) 20      B) 24      C) 25      D) 15
- ```
int f()
{ static int i=0;
int s=1;
s+=i; i++;
return s;
}

main()
{ int i,a=0;
for(i=0;i<5;i++) a+=f();
printf("%d\n",a);
}
```
- 204 下面程序的输出结果是\_\_\_\_      A、110      B、102      C、100      D、105
- ```
static int a[]={1,3,5,7,9};
int y,x,*p;
y=1;
p=&a[1];
for(x=0;x<3;x++) y*=*(p+x);
printf("%d",y);
```
- 205 下面函数的功能是      A) 求字符串的长度      B) 比较两个字符串的大小  
C) 将字符串x复制到字符串y      D) 将字符串x连接到字符串y后面
- ```
int fun1(char * x)
{ char *y=x;
while(*y++);
return(y-x-1);}
```
- 206 下面函数的功能是      A) 求字符串的长度      B) 比较两个字符串的大小  
C) 将字符串s复制到字符串t中      D) 将字符串s接续到字符串t中
- ```
sss(char *s,char *t)
{ while( (*s) && (*t) && ( *t++ == *s++ ) ); return(*s-*t); }
```
- 207 下面能正确进行字符串赋值操作的语句是      A) char s[5]={“ABCDE”};      B) char s[5]={‘a’,‘b’,‘c’,‘d’,‘e’};  
C) char \*s;s=“ABCDEF”      D) char \*s; scanf(“%s”,s);
- 208 下述对C语言字符数组的描述中错误的是      A) 字符数组可以存放字符串      B) 字符数组中的字符串可以整体输入、输出  
C) 可以在赋值语句中通过赋值运算符“=”对字符数组整体赋值      D) 不可以用关系运算符对字符数组中的字符串进行比较
- 209 判断char型变量c1是否为小写字母 正确表达式为:      A) ‘a’ <=c1<= ‘z’      B) (c1>= ‘a’)&&(c1<= ‘z’)  
C) (‘a’ >=c1)|| (‘z’ <= c1)      D) (c1>= ‘a’)&&(c1<= ‘z’)
- 210 若w, x, y, z, m均为int型变量, 则执行下面语句后的m值是:      A) 1      B) 2      C) 3      D) 4
- ```
w= 1; x = 2; y= 3; z=4;
m=(w<y)?w:x;
m= (m<y)? m:y;
m=(m<z)?m:Z;
```
- 211 下面各语句行中, 正确执行字符串操作的语句行是      A) char st[4][5]={“ABCDE”};      B) char s[5]={‘A’,‘B’,‘C’,‘D’,‘E’};  
C) char \*s;s= “ABCDE” ;      D) char \*s; scanf( “%S” ,S);

212 下面程序的输出是\_\_\_\_\_ A) 360 B) 3600 C) 1080 D) 1200

```
int w=3;
main()
{ int w=10; printf("%d\n", fun(5)*w); }
fun(int k)
{ if(k==0) return w;
return(fun(k-1)*k); }
```

B

213 下面程序的输出是\_\_\_\_\_

```
main()
{ char *s="12134211";
int v1=0, v2=0, v3=0, v4=0, k;
for(k=0; s[k]; k++)
switch(s[k])
{ default: v4++;
case '1': v1++;
case '3': v3++;
case '2': v2++; }
printf("v1=%d, v2=%d, v3=%d, v4=%d\n", v1, v2, v3, v4); }
```

A) v1=4, v2=2, v3=1, v4=1      B) v1=4, v2=9, v3=3, v4=1  
C) v1=5, v2=8, v3=6, v4=1      D) v1=8, v2=8, v3=8, v4=8

C

214 下面程序的输出是      A) 123 B) 1239 C) 123456123 D) 123456789

```
main()
{ char ss[10]="12345";
strcat(ss, "6789");
printf("%s\n", ss);
}
```

D

215 下面程序的输出是      A) 0 1 2 3      B) 0 4 0 10      C) 0 4 5 15      D) 1 4 5 15

```
main()
{ enum team {my, your=4, his, her=his+10};
printf("%d %d %d %d\n", my, your, his, her); }
```

C

216 下面程序的输出是      A) 0      B) 1      C) 3      D) 不确定的值

```
main()
{ int x=10, y=3;
printf("%d\n", y=x/y); }
```

C

217 下面程序的输出是      A) -1      B) 0      C) 1      D) 不确定的值

```
main()
{ int x=100, a=10, b=20, ok1=5, ok2=0;
if(a<b)
if(b!=15)
if(! ok1)
x=1;
else
if(ok2)x=10;
x=-1;
printf("%d\n", x); }
```

A

218 下面程序的输出是      A) 1      B) 2      C) 7      D) 10

```
int m=13;
int fun2(int x, int y)
{ int m=3;
return (x*y-m);
}
main()
{ int a=7, b=5;
printf("%d\n", fun2(a, b)/m); }
```

B

219 下面正确的字符常量是:

A) "a"      B) '\\"'      C) 'w'      D) ''

C

220 下面程序的输出是      A) 17      B) 18      C) 23      D) 24

```
main()
{ int x=023;
printf("%d\n", --x); }
```

B

- 221 下面程序的输出是      A) 23      B) 26      C) 33      D) 36
- ```
int aa[3][3]={ {2}, {4}, {6} };
main()
{ int i, * p=&aa[0][0];
for(i=0; i<2;i++)
{ if(i==0)
aa[i][i+1]=*p+1;
else
++p;
printf("%d",*p);}
}
```
- A
- 222 下面程序的输出是      A) 3      B) 4      C) 1      D) 2
- ```
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10},*p=a;
printf("%d\n",*(p+2));}
```
- A
- 223 下面程序的输出是      A) 3      B) 4      C) 6      D) 9
- ```
fun3(int x)
{static int a=3;
a+=x;
return(a);}
main()
{ int k=2, m=1, n;
n=fun3(k);
n=fun3(m);
printf("%d\n",n);}
```
- C
- 224 下面程序的输出是      A) 60      B) 68      C) 99      D) 108
- ```
main()
{ int a[3][4]={ 1,3,5,7, 9,11,13,15, 17,19,21,23};
int (*p)[4]=a, i, j, k=0;
for(i=0; i<3; i++)
for(j=0; j<2; j++)
k=k+*( *(p+i)+j);
printf("%d\n",k);}
```
- A
- 225 下面程序的输出是      A) 741      B) 852      C) 963      D) 875421
- ```
main()
{ int y=9;
for( ; y>0; y--)
{ if(y%3==0)
{ printf("%d", --y);continue;}
}
}
```
- B
- 226 下面程序的输出是
- A) ABCD    B) A    C) D    D) ABCD
- |     |   |   |     |
|-----|---|---|-----|
| BCD | B | C | ABC |
| CD  | C | B | AB  |
| D   | D | A | A   |
- char s[]="ABCD";
- ```
main()
{ char *p;
for(p=s; p<s+4; p++)
printf("%s\n",p);}
```
- A
- 227 下面程序把数组元素中的最大值放入a[0]中。则在if 语句中的条件表达式应该是
- A) p>a    B) \*p>a[0]    C) \*p>\*a[0]    D) \*p[0]> \*a[0]
- ```
main( )
{ int a[10]={6,7,2,9,1,10,5,8,4,3},*p=a, i;
for(i=0;i<10;i++,p++)
if(_____) *a=*p;
printf("%d",*a);
}
```
- B


- 228 下面程序的输出结果是      A) -1    B) 0    C) 1    D) 2
- ```
main()
{ int i=2,p;
p=f(i,i+1);
printf("%d",p);
}
int f(int a,int b)
{int c;
if(a>b) c=1;
else
if(a==b) c=0;
else c=-1;
return c;
}
```
- 229 下面程序的输出结果是      A) 0      B) 1      C) 10      D) 9
- ```
main()
{ int a[ ]={1,2,3,4,5,6,7,8,9,10},*p;
p=a;
printf("%d\n",*p+9);
}
```
- 230 下面程序的输出是    A) a=32768      B) a=32767      C) a=-32767      D) a=-1
- ```
main()
{ unsigned a=32768;
printf("a=%d\n",a);
}
```
- 231 下面程序的输出是    A) abcdefgh      B) hgfdceba      C) dcba hgf      D) hgfdceba
- ```
#include<string.h>
main()
{ char *a="abcdefgh";
fun(a);puts(a); }
fun (char * s)
{ int x=0,y;char c;
for(y=strlen(s)-1;x<y;x++,y--) { c=s[x];s[x]=s[y];s[y]=c;} }
```
- 232 设有程序段
- ```
int k=10;
while (k=0) k= k-1;
```
- 则下面描述中正确的是
- A) while循环执行10次      B) 循环是无限循环  
C) 循环体语句1次也不执行    D) 循环体语句执行一次
- 233 下列程序执行后输出的结果是    A) 3      B) 0      C) 5      D) 4
- ```
f(int a)
{ int b=0; static c=3;
a=c++,b++;
return (a); }
main( )
{ int a=2,i,k;
for(i=0;i<2;i++) k=f(a++);
printf("%d\n",k); }
```
- 234 下列程序执行后输出的结果是    A) 8 4    B) 9 6    C) 9 4    D) 8 5
- ```
int d=1;
fun (int p)
{ int d=5;
d+=p++; printf("%d",d); }
main( )
{ int a=3;
fun(a);
d+=a++; printf("%d\n",d); }
```
- 235 下列程序执行后的输出结果是    A) you&me    B) you    C) me    D) err
- ```
main()
{ char arr[2][4];
strcpy(arr,"you"); strcpy(arr[1],"me");
arr[0][3]='&';
printf("%s\n",arr);
}
```

- 236 下列程序执行后的输出结果是      A) hello    B) hel    C) hlo    D) hlm
- ```

void func1(int i);
void func2(int i);
char st[]="hello,friend!";
void func1(int i)
{ printf("%c",st[i]);
  if(i<3){i+=2;func2(i);}
}
void func2(int i)
{ printf("%c",st[i]);
  if(i<3){i+=2;func1(i);}
}
main()
{ int i=0; func1(i); printf("\n");}

```
- 237 下列程序执行后的输出结果是      A) 6    B) 7    C) 8    D) 9
- ```

void func(int *a,int b[])
{ b[0]=*a+6; }
main()
{ int a,b[5];
  a=0; b[0]=3;
  func(&a,b); printf("%d \n",b[0]);
}

```
- 238 下面程序的输出结果是      A:11, 10    B:11, 11    C:10, 10    D:10, 11 
- ```

int x=10;
int y=x++;
printf("%d,%d", (x++,y),y++);

```
- 239 下列程序的运行结果是      A) a=%2, b=%5    B) a=2, b=5    C) a=d, b=d    D) a=%d, b=%d
- ```

#include <stdio.h>
main()
{ int a=2, c=5;
  printf("a=%d, b=%d\n", a, c); }

```
- 240 请阅读以下程序:
- ```

main()
{float a,b;
  scanf("%f",&a);
  if (a<10.0) b=1.0/a;
  else if((a<0.5)&&(a!=2.0))b= 1.0/(a+2.0);
  else if(a<10.0) b= 1.0/x ;
  else b=10.0;
  printf("%f\n",b);
}

```
- 若运行时输入2.0(回车), 则上面程序的输出结果是:  
A) 0.000000    B) 0.500000    C) 1.000000    D) 0.250000
- 241 以下不正确的if语句形式是:
- A) if(x>y && x!=y);  
B) if(x= =y) x+=y ;  
C) if(x != y) scanf("%d",&x ) else scanf("%d",&y);  
D) if(x<y) {y++;y++;}
- 242 下列运算符中优先级最低的是:    A)?:    B)&&    C)+    D) !=
- 243 设有以下函数:
- ```

f( int a)
{ int b=0;
  static int c = 3;
  b++; c++;
  return(a+b+c);
}

```
- 如果在下面的程序中调用该函数, 则输出结果是
- ```

main()
{ int a = 2, i;
  for(i=0;i<3;i++) printf("%d\n",f(a));
}

```
- A) 7    B) 7    C) 7    D) 7  
8    9    10    7  
9    11    13    7
- 244 设正x、y均为整型变量, 且x=10 y=3, 则以下语句的输出结果是
- ```

printf("%d,%d\n",x--,--y);

```
- A) 10, 3    B) 9, 3    C) 9, 2    D) 10, 2

- 245 执行如下程序后,ab的值为 A) 120 B) 110 C) 100 D) 90  

```
int *var, ab;
ab=100;
var=&ab;
ab=*var+10;
```

B
- 246 设有如下的函数,则函数的类型 A. 与参数x的类型相同 B. 是void C. 是int D. 无法确定  

```
ggg(x)
{float x;
printf("\n%d", x*x);}
```

C
- 247 设有static char str[]="Beijing";则执行  

```
printf("%d\n", strlen(strcpy(str, "China")));
```

后的输出结果为 A) 5 B) 7 C) 12 D) 14 A
- 248 设 int x=1, y=1; 表达式(!x||y-- )的值是 A) 0 B) 1 C) 2 D) -1 B
- 249 若有以下说明:  

```
int a[12]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
char c='a', d, g;
```

则数值为4的表达式是 A) a[g-c] B) a[4] C) a['d'-'c'] D) a['d'-'c'] D
- 250 若有以下说明:  

```
int w[3][4]={ {0, 1}, {2, 4}, {5, 8} };
int (*p)[4]=w;
```

则数值为4的表达式是 A) \*w[1]+1 B) p++,\*(p+1) C) w[2][2] D) p[1][1] D
- 251 若有以下说明和语句,请选出哪个是对c数组元素的正确引用  

```
int c[4][5], (*cp)[5];
cp=c;
```

A) cp+1 B) \*(cp+3) C) \*(cp+1)+3 D) (\*(cp+2)) D
- 252 若有以下定义和语句,则输出结果是 A) 4 B) 5 C) 9 D) 10  

```
char *s1="12345", *s2="1234";
printf("%d\n", strlen(strcpy(s1, s2)));
```

A
- 253 若有以下定义和语句,则输出结果是: A) 8, 16, 10 B) 10, 10, 10 C) 8, 8, 10 D) 8, 10, 10  

```
int u=010, v=0x10, w=10;
printf("%d, %d, %d\n", u, v, w);
```

A
- 254 若有以下定义:char s[20]="programming", \*ps=s;  
则不能代表字符o的表达式是 A) ps+2 B) s[2] C) ps[2] D) ps+=2, \*ps A
- 255 若有以下的说明和语句:  

```
main()
{int t[3][2], *pt[3], k;
for(k=0; k<3; k++) pt[k]=t[k];
}
```

则以一选项中能正确表示t数组元素地址的表达式是  
A) &t[3][2] B) \*pt[0][0] C) \*(pt+1) D) &pt[2] C
- 256 若有以下定义:float x;int a, b;则正确的switch 语句是  
A) switch(x) B) switch(x)  

```
{ case 1.0:printf("*\n"); { case 1, 2:printf("*\n");
case 2.0:printf("**\n"); case 3 :printf("**\n");
}
```

C) switch(a+b) D) switch (a+b);  

```
{ case 1 :printf("\n"); { case 1:printf("*\n");
case 1+2:printf("**\n"); case 2:printf("**\n");
}
```

C
- 257 请阅读以下程序:  

```
main()
{int a=5, b=0, c=0;
if(a=b+c) printf("***\n ");
else printf("$$$ \n");
}
```

以上程序 :  
A) 有语法错不能通过编译 B) 可以通过编译但不能通过连接  
C) 输出\*\*\*D) 输出\$\$\$ D
- 258 若有以下程序片段,则输出结果是 A) 3 B) 4 C) 6 D) 12  

```
char str[]="ab\n012\\\"";
printf("%d", strlen(str));
```

C
- 259 若有说明, double \*p, a;则能通过scanf语句正确给输入项读入数据的程序段是  
A) \*p=&a; scanf("%lf", p); B) p=(double \*)malloc(8);scanf("%f", p);  
C) p=&a;scanf("%lf", a); D) p=&a; scanf("%le", p); D
- 260 若有说明和语句,则表达式a++的值是 A) 7 B) 6 C) 5 D) 4  

```
int a=5;
a++;
```

C

- 261 若有下面的程序片段:
- ```
int a[12]={0}, *p[3], **pp, i;
for(i=0; i<3; i++)
    p[i]=&a[i*4];
pp=p;
```
- 则对数组元素的错误引用是 A) pp[0][1] B) a[10] C) p[3][1] D) \*(\*(p+2)+2)
- 262 若有下面的说明和定义, 则sizeof(struct aa) 的值是 A) 2 B) 4 C) 8 D) 14
- ```
struct aa
{ int r1; double r2; float r3; }
```
- 263 若有定义: float w; int a, b; 则合法的switch语句是
- A) switch(w) B) switch(a);
- ```
{ case 1.0: printf("*\n"); case 1 printf("*\n");
  case 2.0: printf("**\n"); case 2 printf("**\n");
}
```
- C) switch(b); D) switch(a+b)
- ```
{ case 1: printf("*\n"); case 1: printf("*\n");
  default: printf("\n"); case 2: printf("**\n");
  case 1+2: printf("**\n"); default: printf("\n");
}
```
- 264 若已定义: int a[ ]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, \*p=a, i;
- 其中 0≤i≤9, 则对a数组元素不正确的引用是 A) a[p-a] B) \*(&a[i]) C) p[i] D) a[10]
- 265 若变量a、i已正确定义, 且i已正确赋值, 合法的语句是
- A) a==i B) ++i; C) a=a++=5; D) a=int(i);
- 266 若k是int型变量, 且有下列的程序片段:
- ```
k=-3
if(k<=0) printf("####")
else printf("&&&&");
```
- 输出结果是: A) ##### B) &&&& C) #####&&&& D) 有语法错误, 无输出结果
- 267 若x是int型变量, 且有下列的程序片段:
- ```
for(x=3; x<6; x++) printf((x%2)?("**%d"): ("###d\n"), x);
```
- 上面程序片段的输出结果是
- A) \*\*3 B) ##3 C) ##3 D) \*\*3##4
- ```
##4 **4 **4##5 **5
**5 ##5
```
- 268 若x是整型变量, pb是基类型为整型的指针变量, 则正确的赋值表达式是\_\_\_\_\_
- A) pb=&x B) pb=x; C) \*pb=&x; D) \*pb=\*x
- 269 若I, j已定义为int类型, 则以下程序段中内循环体的总的执行次数是
- A) 20 B) 25 C) 24 D) 30
- ```
for (i=5; i; i--)
for(j=0; j<4; j++) {...}
```
- 270 请选出以下语句的输出结果 A) 5 B) 14 C) 8 D) 输出项不合法, 无正常输出
- ```
printf("%d\n", strlen("\t\t\065\xff\n"));
```
- 271 请选出正确的程序段
- A) int \*p; B) int \*s, k;
- ```
scanf("%d", p); *s=100;
... ..
```
- C) int \*s, k; D) int \*s, k;
- ```
char *p, c; char *p, e;
s=&k; s=&k;
p=&c; p=&c;
*p='a'; s=p;
... *s=1;
... ..
```
- 272 若a、b、c1、c2、x、y、均是整型变量, 正确的switch语句是
- A) swich(a+b); B) switch(a\*a+b\*b)
- ```
{ case 1:y=a+b; break; {case 3:
  case 0:y=a-b; break; case 1:y=a+b; break;
  case 3:y=b-a; break;
}
```
- C) switch a D) switch(a-b)
- ```
{ case c1 :y=a-b; break {default:y=a*b; break
case c2: x=a*d; break case 3:case 4:x=a+b; break
default:x=a+b; case 10:case 11:y=a-b; break;
}
```



273 请选出以下程序的输出结果

```
#include<stdio.h>
fun(s, n1, n2)
int *s, n1, n2;
{ int i, j, t;
i=n1; j=n2;
while(i<j)
{ t= *(s+i); *(s+i)= *(s+j); *(s+j)=t;
i++; j--;
}
}
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,0}, i, *p=a;
fun(p, 0, 3); fun(p, 4, 9); fun(p, 0, 9);
for(i=0; i<10; i++) printf("%d", *(a+i));
printf("\n");
}
```

A) 0 9 8 7 6 5 4 3 2 1      B) 4 3 2 1 0 9 8 7 6 5

C) 5 6 7 8 9 0 1 2 3 4      D) 0 9 8 7 6 5 1 2 3 4

274 以下程序的输出结果是: A) 7      B) 8      C) 9      D) 10

```
main()
{ int a, b;
for (a=1,b=1;a<=100;a++)
{ if(b>=20) break;
if(b%3==1)
{b+=3;
continue;
}
b-=5;
}
printf("%d\n",a);
}
```

275 请选出以下程序的输出结果      A) 1 2 3 4 B) 4 3 2 1 C) 0 0 0 0 D) 4 4 4 4

```
main()
{ int a[ ]={ 1,2,3,4 }, i;
int x=0;
for(i=0; i<4; i++)
{ sub(a, x); printf("%d", x);}
printf("\n");
}
sub(int *s, int y)
{ static int t=3;
y=s[t]; t--;
}
```

276 请选出以下程序的输出结果

```
sub(int x, int y, int *z)
{ *z=y-x; }
main()
{ int a, b, c;
sub(10, 5, &a); sub(7, a, &b); sub(a, b, &c);
printf("%d,%d,%d\n", a, b, c);
}
```

A) 5, 2, 3 B) -5, -12, -7 C) -5, -12, -17 D) 5, -2, -7

277 请选出以下程序段的输出结果

```
main()
{ char s1[10], s2[10], s3[10], s4[10];
  scanf("%s%s", s1, s2); gets(s3); gets(s4);
  puts(s1); puts(s2); puts(s3); puts(s4);
}
```

输入数据如下: (此处<CR>代表回车符)

```
aaaa bbbb<CR>
cccc dddd<CR>
```

- A) aaaa bbbb  
B) aaaa bbbb  
C) aaaa bbbb  
D) aaaa bbbb

A

278 putchar函数可以向终端输出1个:

- A) 整型变量表达式  
B) 实型变量值  
C) 字符串  
D) 字符或字符型变量值

D

279 请读程序:

```
void fun(float *p1, float *p2, float *s)
{ s=( float * )malloc( 1, sizeof(float));
  *s=*p1+ *(p2++);
}

main()
{ float a[2]={1.1, 2.2}, b[2]={10.0, 20.0}, *s=a;
  fun(a, b, s);
  printf("%f\n", *s);
}
```

上面程序的输出结果是

- A) 11.100000 B) 12.100000  
C) 21.100000 D) 1.100000

D

280 请读程序:

```
#include<stdio.h>
#include<string.h>
void fun( char *w, int m)
{ char s, *p1, *p2;
  p1=w; p2=w+m-1;
  while (p1<p2)
  { s=*p1++; *p1=*p2--; *p2=s;}
}

main()
{ char a[]="ABCDEFGH";
  fun( a, strlen(a));
  puts(a);
}
```

上面程序的输出结果是

- A) GFEDCBA B) AGADAGA C) AGAAGAG D) GAGGAGA

C

281 以下程序的输出结果是: A) 正数 B) 负数 C) 零 D) 不确定的值

```
#include<stdio.h>
#include<string.h>
main()
{char *s1="AbCdEf", *s2="aB";
  s1++; s2++;
  printf("%d\n", strcmp(s1, s2));
}
```

A

282 两次运行下面的程序, 如果从键盘上分别输入6和4, 则输出结果是

- A) 7和5 B) 6和3 C) 7和4 D) 6和4
- ```
main( )
{ int x;
  scanf("%d",&x);
  if(x ++ >5) printf("%d", x);
  else printf("%d\n", x --); }
```

A

283 能正确表示a和b同时为正或同时为负的逻辑表达式是

- A)  $(a \geq 0 \parallel b \geq 0) \&\& (a < 0 \parallel b < 0)$     B)  $(a \geq 0 \&\& b \geq 0) \&\& (a < 0 \&\& b < 0)$   
C)  $(a+b > 0) \&\& (a+b \leq 0)$     D)  $a*b > 0$

D

284 请读程序:

```
#include<math.h>
#include<stdio.h>
main()
{ float x,y,z;
  scanf("%f%f",&x,&y);
  z=x/y;
  while(1)
  { if(fabs(z)>1.0)
    { x=y; y=z; z=x/y;
    }
    else break;
  }
  printf("%f\n",y);
}
```

B

若运行时从键盘上输入3.6 2.4<CR>(<CR>表示回车), 则输出的结果是

- A) 1.500000    B) 1.600000    C) 2.000000    D) 2.400000

285 请读程序:

```
#include<stdio.h>
#include<string.h>
void fun( char *s)
{ char a[10];
  strcpy( a, "STRING");
  s=a;
}
main()
{ char *p;
  fun(p);
  printf("%s\n",p);
}
```

D

上面程序的输出结果是( ☐ 表示空格)

- A) STRING ☐ ☐ ☐ ☐    B) STRING    C) STRING ☐ ☐ ☐    D) 不确定的

286 请读程序:

```
#include<stdio.h>
char fun(char *c)
{if( *c<='Z' && *c>='A') *c--='A'-'a';
  return *c;
}
main()
{ char s[81], *p=s;
  gets(s);
  while(*p)
  { *p=fun(p); putchar(*p); p++;}
  putchar('\n');
}
```

B

若从键盘上输入OPEN THE DOOR<CR>(<CR>表示回车), 则上面程序的输出结果是

- A) oPEN tHE dOOR    B) open the door    C) OPEN THE DOOR    D) Open The Door

287 以下程序的输出结果是:    A)6    B)8    C)4    D)2

```
#include<stdio.h>
int a[]={2,4,6,8};
main()
{ int i;
  int *p=a;
  for(i=0;i<4;i++) a[i]=*p++;
  printf("%d\n",a[2]);
}
```

A

288 请读程序:

```
#include<stdio.h>
main()
{ int mun=0;
  while(num<=2)
  { num++; printf("%d\n",num);}
}
```

输出结果是 A) 1      B) 1      C) 1      D) 1  
                                2                  2                  2  
                                                        3                  3  
                                                                                4

C

289 请读程序:

```
#include<stdio.h>
main()
{ int x=1, y=0, a=0, b=0;
  switch(x)
  { case 1:
    switch(y)
    { case 0: a++;break;
      case 1: b++;break;
    }
    case 2:
      a++; b++; break;
  }
  printf("a=%d, b=%d\n",a,b);
}
```

上面程序的输出结果是    A) a=2, b=1    B) a=1, b=1    C) a=1, b=0    D) a=2, b=2

A

290 请读程序:

```
#include<stdio.h>
void fun( int *s)
{static int j=0;
  do
  { s[j]+=s[j+1];
    while(++j<2);
  }
main()
{ int k, a[10]={1,2,3,4,5};
  for(k=1; k<3; k++) fun(a);
  for(k=0; k<5; k++) printf("%d", a[k]);
}
```

程序的输出结果是:    A) 34756    B) 23445    C) 35745    D) 1234

291 假设在程序中a, b, c均被定义成整型, 并且已赋大于1的值, 则下列能正确表示代数式 的表达式是

A) 1/a\*b\*c                  B) 1/(a\*b\*c)                  C) 1/a/b/(float)c                  D) 1.0/a/b/c

D

292 假定x和y为 double型, 则表达式x=2, y=x+3/2的值是

A) 3.500000    B) 3    C) 2.000000    D) 3.000000

D

293 下面程序段的运行结果是:

```
x = y= 0;
while(x<15)    y++, x+=++y;
printf("%d,%d", y, x);
```

A) 20, 7    B) 6, 12    c) 20, 8    D) 8, 20

D

294 合法的数组定义是

A) int a[]="string"; B) int a[5]={0, 1, 2, 3, 4, 5};  
C) char s="string"; D) char a[]={0, 1, 2, 3, 4, 5};

D

295 假定a和b为int型变量, 则执行以下语句后b的值为    A) 9    B) -2    C) -1    D) 8

a=1; b=10;

do

{ b-=a; a++; }while (b--<0) ;

D

296 当执行以下程序段时

A 循环体将执行一次      B 循环体将执行两次  
C 循环体将执行无限次    D 系统将提示有语法错误

A

x=-1;

do { x=x\*x;} while (! x) ;

297 当执行下面程序且输入:ABC时,输出的结果是 A)ABC B)ABC9 C)123456ABC D) ABC456789

```
#include<stdio.h>
#include<string.h>
main()
```

```
{ char ss[10]="12345";
  gets(ss);printf("%s\n",ss); }
```

298 当c的值不为0时,在下列选项中能正确将c的值赋给变量a、b的是\_\_\_\_\_

A) c=b=a; B) (a=c) || (b=c); C) (a=c) && (b=c); D) a=c=b;

299 不能把字符串:Hello!赋给数组b的语句是

A) char b[10]={'H','e','l','l','o','!'}; B) char b[10];b="Hello!";  
C) char b[10];strcpy(b,"Hello!"); D) char b[10]="Hello!";

300 t为int类型,进入下面的循环之前,t的值为0,则以下叙述中正确的是

```
while( t=1 )
{ ..... }
```

A) 循环控制表达式的值为0 B) 循环控制表达式的值为1  
C) 循环控制表达式不合法 D) 以上说法都不对

301 变量a所占内存字节数是 A) 4 B) 5 C) 6 D) 8

```
union U
{ char st[4];
  int i;
};
struct A
{ int c;
  union U u;
}a;
```

302 当运行以下程序时,从键盘输入:AhaMA Aha<CR>(<CR>表示回车),则下面程序的运行结果是

```
main()
{ char s[80],c='a';
  int i=0;
  scanf("%s",s);
  while(s[i]!='\0')
  { if(s[i]==c) s[i]=s[i]-32;
    else if(s[i]==c-32) s[i]=s[i]+32;
    i++; }
  puts(s); }
```

A) ahAma B) AhAMa C) AhAma ahA D) ahAMa ahA

303 下面程序的运行结果是\_\_\_\_\_。

```
#include<stdio.h>
#include<string.h>
main()
{ char a[80]="AB",b[80]="LMNP"; int i=0;
  strcat(a,b);
  while(a[i++]!='\0') b[i]=a[i];
  puts(b); }
```

A) LB B) ABLMNP C) AB D) LBLMNP

304 当\_\_\_\_,可以不指定数组长度。

(A) 对静态数组赋初值 (B) 对动态数组赋初值  
(C) 只给一部分元素赋值 (D) 对全部数组元素赋初值

305 以下程序的输出结果是\_\_\_\_\_ A、5 B、6 C、23 D、26

```
i=5;
i=++i*4+2;
```

306 以下程序的输出结果是\_\_\_\_\_ A、5 B、6 C、23 D、26

```
i=5;
i=i++*4+2;
```

307 以下程序的输出结果是 A) 1, 1, 1 B) 1, 0, 0 C) 1, 1, 0 D) 1, 0, 1

```
main()
{int x,y,z;
  x=y=z=0;
  ++x||++y||++z;
  printf(" %d,%d,%d\n",x,y,z);
}
```

308 C语言中函数能否嵌套调用和递归调用?

A、二者均不可 B、前者可,后者不可 C、前者不可,后者可 D、二者均可

309 C语言联合类型在任何给定时刻\_\_\_\_\_。

A. 所有成员一直驻留在结构中。 B. 只能有一个成员驻留在结构中。  
C. 部分成员驻留在结构中。 D. 没有成员驻留在结构中。

310 #include<stdio.h>

```
main()
{ int a,b,c,d;
  scanf("%d%d",&a,&b);
  c=gbs(a,b);
  d=gys(a,b);
  printf("a=%db=%d gbs=%d gys=%d\n",a,b,c,d);
}
gbs(a,b)
int a,b;
{ int c;
  a=a>0?a:-a;
  b=b>0?b:-b;
  c=a*b;
  if(c==0) return(0);
  while(a!=b) if(a>b) a-=b;
  else b-=a;
  return(c/a);
}
gys(a,b)
int a,b;
{ if(a*b==0) return(0);
  a=a>0?a:-a;
  b=b>0?b:-b;
  while(a!=b) if(a>b) a-=b;else b-=a;
  return(a);
}
```

请写出上面程序的运行结果:

A、 a=30 b=63 gbs=189 gys=9      B、 a=27 b=63 gbs=190 gys=10  
C、 a=27 b=63 gbs=189 gys=9      D、 a=27 b=63 gbs=188 gys=10

311 请读程序:

```
main()
{ int *p,x;
  x=10;
  *p=x;
  printf("%d\n",*p);
}
```

请判断上面程序\_\_\_\_\_

A、 错误:\*p=x;      B、 错误:\*p=x;  
printf("%d\n"),\*p);      printf("%d\n"),\*p);  
正确:p=&x;      正确:p=\*\*x;  
printf("%d\n",\*p);      printf("%d\n",p);  
C、 错误:int \*p,x;      D、 错误:\*p=x;  
正确:int \*(p),\*(x);      正确:p=&x;

312 根据下面的定义,能打印出字母M的语句是\_\_\_\_\_

A) printf("%c\n",class[3].name);      D) printf("%c\n",class[2].name[0]);  
B) printf("%c\n",class[3].name[1]);      C) printf("%c\n",class[2].name[1]);  
struct person { char name[9]; int age;};  
struct person class[10]={"John",17,"Paul",19,"Mary",18,"Adam",16};

313 下面程序的功能是将字符串s中所有的字符c删除。请选择填空。

```
main()
{ char s[80]; int i,j;
  gets(s);
  for(i=j=0;s[i]!='\0';i++)
    if(s[i]!='c') 【1】
  puts(s); }
```

【1】 A) s[j++]=s[i]      B) s[++j]=s[i];      C) s[j]=s[i];j++;      D) s[j]=s[i];

314 假定下列程序的可执行文件名为prg.exe,则在该程序所在的子目录下输入命令行: prg hello good<回车>后,程序的输出结果是      A) hello good      B) hg      C) hel      D) hellogood

```
main() (int argc, char *argv[])
{ int i;
  if(argc<=0) return;
  for(i=1;i<argc;i++) printf("%c", *argv[i]);
}
```

315 判断字符串s1是否大于字符串s2应当使用\_\_\_\_\_。

A) if(s1>s2)      B) if(strcmp(s1,s2))      C) if(strcmp(s2,s1)>0)      D) if(strcmp(s1,s2)>0)

- 316 利用数组写出一个与int \*p[4]等价的说明  
A、int &p[4];      B、int p[4];      C、int \*\*p[4];      D、int \*(\*p[4]);      B
- 317 请选出以下程序段的输出结果    A) 15 B) 100 C) 10 D) 150  
#include<stdio.h>  
#define MIN(x,y) (x)<(y)? (x):(y)  
main()  
{ int i,j,k;  
i=10; j=15;  
k=10\*MIN(i,j);  
printf("%d\n",k);  
}  
A
- 318 若a=5,b=3,c=4,逻辑表达式!(a+b)&&!c的值为    (A) 0    (B) -1    (C) 1    (D) 2      A
- 319 设以下变量均为int类型,则值不等于7的表达式是  
A) (x= y= 6, x+y,x+1)      B) (x= y= 6,x+y,y+1)      C  
C) (x= 6,x+1,y= 6,x+y)      D) (y=6,y+1,x = y,x+1)
- 320 若程序中有下面的说明和定义,则会发生的情况是  
struct abc  
{int x;char y;}  
struct abc s1,s2;  
A) 编译进错      B) 程序将顺利编译`连接`执行  
C) 能顺利通过编译`连接`但不能执行      D) 能顺利通过编译`但连接出错
- 321 以下不能对二维数组a进行正确初始化的语句是\_\_\_\_\_  
A) int a[2][3]={0};  
B) int a[][3]={1,2,3},{4,5,6}};  
C) int a[2][4]={1,2,3},{4,5}{6}};  
D) int a[][3]={1,0,1},{},{1,1}};  
C
- 322 有字符数组a[80]和b[80],则正确的输出形式是\_\_\_\_\_  
A) puts(a,b);      B) printf("%s,%s,a[],b[]);  
C) putchar(a,b);      D) puts(a),puts(b);      D
- 323 若给定条件表达式(M)?(a++):(a--)则其中表达式M和\_\_\_\_\_等价。  
A、(M==0)    B、(M==1)    C、(M!=0)    D、(M!=0)      C
- 324 若已定义:int a[9],\*p=a;并在以后的语句中未改变p的值,不能表示a[1] 地址的表达式是  
A) p+1    B) a+1    C) a++    D) ++p      C
- 325 判断字符串a和b是否相等,应当使用\_\_\_\_\_  
A) if(a==b)      B) if(a=b)  
C) if(strcmp(a,b))      D) if(!strcmp(a,b))      D
- 326 以下程序的输出结果是\_\_\_\_    A、1, 1, 1    B、1, 0, 0    C、1, 1, 0    D、1, 0, 1  
main()  
{int x,y,z;  
x=y=z=0;  
++x&&++y||++z;  
printf( "%d,%d,%d\n" , x,y,z);  
}  
C
- 327 有下面的程序段  
char a[3],b[]="china"; a=b;  
printf("%s",a);  
则\_\_\_\_\_  
A) 运行后将输出Chm      B) 运行后将输出Ch  
C) 运行后将输出Chi      D) 编译出错      D
- 328 若有如下定义和语句,则输出结果是:    A) 10, 20    B) 10, 10    C) 20, 10    D) 20, 20  
int \*\*pp,\*p,a=10,b=20;pp=&p;p=&a;p=&b;printf("%d\n",\*p,\*\*pp);      D
- 329 若有说明语句:int a,b,c,\*d=&c;则能正确从键盘读入三个整数分别赋给变量a,b,c的语句是  
A) scanf("%d%d%d",&a,&b,d);      B) scanf("%d%d%d",&a,&b,&d);      A  
C) scanf("%d%d%d",a,b,d);      D) scanf("%d%d%d",a,b,\*d);
- 330 以下正确的定义语句是\_\_\_\_\_  
A) int a[1][4]={1,2,3,4,5};  
B) float x[3][]={{1},{2},{3}};  
C) long b[2][3]={1},{1,2},{1,2,3}};  
D) double y[][3]={0};      D
- 331 若有说明:int i, j=7, \*p=&i;, 则与i=j;等价的语句是  
A) i= \*p; B) \*p=&j; C) i=&j; D) i=\* \*p;      B
- 332 若有说明:long \*p, a;则不能通过scanf语句正确给输入项读入数据的程序段是  
A) \*p=&a; scanf("%ld", p);      B) p=(long \*)malloc(8); scanf("%ld", p);  
C) scanf("%ld", p=&a);      D) scanf("%ld", &a);      A
- 333 若有以下的定义: int a[]={1,2,3,4,5,6,7,88,9,10}, \*p=a;则值为3的表式是  
A) p+=2, \*(p++)    B) p+=2,\*++p    C) p+=3, \*kp++    D) p+=2, ++\*p      A



- 334 若有以下的说明, 则对初值中字符'a'的引用方式为\_\_\_\_\_。  

```
static struct
{
    char ch;
    double x;
    char a[];
}c[2][2]={{'a', 3.5, "bc"}, {'c', 4.5, "de"}, {'m', 8.6, "abc"}}};
```

A、c.ch B、c[0][0].ch C、c[1][1].ch D、a[0]
- 335 若有以下的说明和语句, 则在执行for语句后, \*(pt+1)+2表示的数组元素是  
A) t[2][0] B) t[2][2] C) t[1][2] D) t[2][1]  

```
int t[3][3], *pt[3], k;
for (k=0;k<3;k++) pt[k]=&t[k][0];
```

C
- 336 若有下面的说明和定义, 则sizeof(struct test)的值是\_\_\_\_\_ A) 12 B) 16 C) 14 D) 9  

```
struct test
{
    int m1; char m2; float m3;
    union uu {char u1[5]; int u2[2];} ua;
} myaa;
```

A
- 337 若有以下定义, 则语句( )是正确的。  
A: s=5 B: a={2, 'a' 1.2} C: printf("%d\n", a) D: n=a  

```
union data
{int i;
 char c;
 float f;
}a;
int n;
```

C
- 338 若有以下定义和语句: int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, \*p=a;  
则不能表示a数组元素的表达式是 A) \*p B) a[10] C) \*a D) a[p-a] B
- 339 若有以下定义和语句: double r=99, \*p=&r; \*p=r;  
则以下正确的叙述是  
A) 以下两处的\*p含义相同, 都说明给指针变量p赋值  
B) 在"double r=99, \*p=&r;"中, 把r的地址赋值给了p所指的存储单元  
C) 语句"\*p=r;"把变量r的值赋给指针变量p D) 语句"\*p=r;"取变量r的值放回r中 D
- 340 若有以下定义和语句:  

```
int w[2][3], (*pw)[3]; pw=w;
```

则对w数组元素非法引用是 A) \*(w[0]+2) B) \*(pw+1)[2] C) pw[0][0] D) \*(pw[1]+2) B
- 341 若有以下说明和定义, 在必要的赋值之后, 对fun函数的正确调用语句是  

```
fun(int *c){ }
main()
{ int (*a)()=fun, *b(), w[10], c;
  ;
}
```

A) a=a(w); B) (\*a)(&c); C) b=\*b(w); D) fun (b); B
- 342 若有以下说明:  

```
int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=a;
```

则数值为6的表达式是  
A) \*p+6 B) \*(p+6) C) \*p+=5 D) p+5 C
- 343 若指针p已正确定义, 要使p指向两个连续的整型动态存储单元, 不正确的语句是  
A) p=2\*(int\*)malloc(sizeof(int)); B) p=(int\*)malloc(2\*sizeof(int));  
C) p=(int\*)malloc(2\*2); D) p=(int\*)malloc(2, sizeof(int)); A
- 344 设已有定义: char \*st="how are you"; 下列程序段中正确的是  
A) char a[11], \*p; strcpy(p=a+1, &st[4]); B) char a[11]; strcpy(++a, st);  
C) char a[11]; strcpy(a, st); D) char a[], \*p; strcpy(p=&a[1], st+2); A
- 345 如下程序的输出结果是 A) 运行错误 B) 100 C) i的地址 D) j的地址  

```
#include<stdio.h>
main()
{ int **k, *j, i=100;
  j=&i; k=&j;
  printf("%d\n", **k);
}
```

B
- 346 设一整型(int)变量占用2个字节则下述共同体变量x所占内存字节数为\_\_\_\_\_。  

```
union exp
{ int i;
  float j;
  double k;
}x;
```

A、14个 B、7个 C、8个 D、随机而定 C

- 347 设x的值为4, 则表达式(++x)+(++x)+(++x)的值是 A:12; B:18; C:16; D:21 **B**
- 348 设p1和p2是指向同一个int型一维数组的指针变量,k为int型变量, 则不能正确执行的语句是  
A) k=\*p1\*p2; B) p2=k; C) p1=p2; D) k=\*p1 \* (\*p2); **B**
- 349 设x、y、t均为int型变量, 则执行语句:x=y=3;t=++x||++y;后, y的值为  
A) 不定值 B) 4 C) 3 D) 1 **C**
- 350 定义如下变量和数组, 则下面语句的输出结果是:  
int k; int a[3][3]={1,2,3,4,5,6,7,8,9};  
for(k=0;k<3;k++) printf("%d",a[k][2-k]);  
A) 3 5 7 B) 3 6 9 C) 1 5 9 D) 1 4 7 **A**
- 351 设ch是char型变量, 其值为A, 则下面的表达式的值是 A)A B)a C)Z D)z  
ch=(ch>='A' &&ch<='Z')?(ch+32):ch **B**
- 352 设有如下定义, 若有p=&data;, 则对data中的a域的正确引用是  
struct sk  
{int a;float b;}data,\*p;  
A) (\*p).data.a B) (\*p).a C) p->data.a D) p.data.a **B**
- 353 设有如下定义: int (\*ptr)\*(); 则以下叙述中正确的是  
A) ptr是指向一维数组的指针变量 B) ptr是指向int型数据的指针变量  
C) ptr是指向函数的指针, 该函数返回一个int型数据  
D) ptr是一个函数名, 该函数的返回值是指向int型数据的指针 **C**
- 354 下列程序段不是死循环的是  
A) int i=100;  
while(1)  
{ i=i%100+1;  
if(i>100) break; }  
B) for(;;);  
C) int k=0;  
do{ ++k; } while(k>=0) ;  
D) int s=36;  
while(s); --s; **C**
- 355 设有如下枚举类型定义:  
enum language { Basic=3,Assembly,Ada=100,COBOL,Fortran};  
枚举量Fortran的值为 A) 4 B) 7 C) 102 D) 103 **C**
- 356 设有如下函数定义:  
int f(char \*s)  
{ char \*p=s;  
while(\*p!='\0') p++;  
return(p-s); }  
如果在主程序中用下面的语句调用上述函数, 则输出结果为 A) 3 B) 6 C) 8 D) 0  
printf("%d\n",f("goodbey!")); **C**
- 357 设有说明int (\*ptr)[m];其中的标识符ptr是( )。  
(A) M个指向整型变量的指针 (B) 指向M个整型变量的函数指针  
(C) 一个指向具有M个整型元素的一维数组的指针  
(D) 具有M个指针元素的一维指针数组, 每个元素都只能指向整型量 **C**
- 358 为表示关系 $x \geq y \geq z$ , 应使用C语言表达式  
A) (x>=y)&&(y>=z) B) (x>=y)AND(y>=z) C) (x>=y)>=z D) (x>=y)&(y>=z) **A**
- 359 设有以下定义:  
typedef union{ int k[5];char c;} DATE;  
struct date { int cat;DATE cow;double dog;}too;  
DATE max;  
则下列语句的执行结果是 A) 25 B) 30 C) 18 D) 8  
printf("%d",sizeof(struct date)+sizeof(max)); **B**
- 360 设有以下说明语句, 则下面叙述中正确的是  
typedef struct  
{ int n;  
char ch[8];  
}PER;  
A) PER 是结构体变量名 B) PER是结构体类型名  
C) typedef struct 是结构体类型 D) struct 是结构体类型名 **B**
- 361 设有以下说明语句, 则下面的叙述中不正确的是  
struct ex  
{ int x ; float y; char z ;} example;  
A) struct结构体类型的关键字 B) example是结构体类型名  
C) x,y,z都是结构体成员名 D) struct ex是结构体类型 **B**

- 362 下列程序的输出结果是 A) 4 4 B) 2 2 C) 2 4 D) 4 6
- ```
main( )
{ int a[5]={2,4,6,8,10}, *p, **k;
p=a; k=&p;
printf("%d",*(p++)); printf("%d\n",**k); }
```
- C
- 363 下列程序的输出结果是 A) xyabcAB B) abcABz C) Ababcz D) xycdBCD
- ```
main( )
{ char*p1,*p2, str[50]="xyz";
p1="abcd";
p2="ABCD";
strcpy(str+2, strcat(p1+2, p2+1));
printf("%s", str); }
```
- D
- 364 下列程序的输出结果是 A) 4 B) 6 C) 8 D) 10
- ```
int b=2;
int func(int *a)
{ b += *a; return(b);}
main()
{ int a=2, res=2;
res += func(&a);
printf("%d \n", res);
}
```
- B
- 365 下列程序的输出结果是 A) 5 B) 6 C) 7 D) 8
- ```
struct abc
{ int a, b, c; };
main()
{ struct abc s[2]={ {1,2,3}, {4,5,6} }; int t;
t=s[0].a+s[1].b;
printf("%d \n", t);
}
```
- B
- 366 下列程序的输出结果是 A) 非法 B) a[4]的地址 C) 5 D) 3
- ```
main()
{ char a[10]={9,8,7,6,5,4,3,2,1,0}, *p=a+5;
printf("%d", *--p);
}
```
- C
- 367 下列程序的运行结果是 A) 6 3 B) 3 6 C) 编译出错 D) 0 0
- ```
void fun(int *a, int *b)
{ int *k;
k=a; a=b; b=k;
}
main()
{ int a=3, b=6, *x=&a, *y=&b;
fun(x, y);
printf("%d %d", a, b);
}
```
- B
- 368 请读程序:
- ```
int x, y;
for(x=1; x<5; x+=2)
{ for(y=1; y<5; y++)
printf("%3d", x*y);
if(y==5) printf("\n");
}
```
- A、 1 2 3 4  
0 0 0 0  
B、 0 1 3 5  
0 0 1 2  
C、 1 2 3 4  
3 6 9 12  
D、 1 5 2 8  
0 1 0 1
- C
- 369 下列程序执行后的输出结果是 A) 6 B) 8 C) 10 D) 12
- ```
#define MA(x) x*(x-1)
main()
{ int a=1, b=2; printf("%d \n", MA(1+a+b)); }
```
- B
- 370 下列程序执行后的输出结果是 A) -32767 B) FFFE C) -1 D) -32768
- ```
main()
{ char x=0xFFFF; printf("%d \n", x--); }
```
- C

- 371 下列分支语句合法的是( )。  
 (A) if(a>c) m=a else m=c (B) if a>c if a>b) m=a;  
 (C) if (a>b && a>c) m=a; (D) case 6,7:printf("reset\n"); C
- 372 下面程序的输出结果为  

```

struct st
{ int x;
  int *y;
} *p;
int dt[4]={10,20,30,40};
struct st aa[4]={ 50,&dt[0],60,&dt[1],
70,&dt[2],80,&dt[3] };
main()
{ p=aa;
  printf("%d\n", ++p->x );
  printf("%d\n", (++p)->x);
  printf("%d\n", ++( *p->y));
}

```

 A) 10    B) 50    C) 51    D) 60  
       20       60       60       70  
       20       21       21       31 C
- 373 阅读以下程序,当输入数据的形式为25,13,10<CR>正确的输出结果为:  

```

main()
{int x,y,z
  scanf("%d%d%d",&x,&y,&z );
  printf("x+y+z=%d\n ,x+y+z); }

```

 A)x+y+z=48    B)x +y+z=35    C)x +z=35    D)不确定值 D
- 374 下面程序的输出是    A) 32    B) 16    C) 8    D) 24  

```

typedef union { long x[2];
int y[4];
char z[8];
} MYTYPE;
MYTYPE them;
main()
{ printf("%d\n", sizeof(them));}

```

 C
- 375 下面程序的输出是    A) 0    B) 1    C) 3    D) 6  

```

main()
{ struct cmplx { int x; int y; } cnum[2]={1,3,
2,7};
printf("%d\n",cnum[0].y /cnum[0].x * cnum[1].x);}

```

 D
- 376 以下描述中正确的是:  
 A) 由于do-while循环中循环体语句只能是一条可执行语句,所以循环体内不能使用复合语句  
 B) do-while循环由do开始,用while结束,在while(表达式)后面不能写分号  
 C) 在do-while循环体中,一定要有能使while后表达式值变为零("假")的操作  
 D) do-while循环中,根据情况可以省略while C
- 377 下面程序的运行结果是 \_\_\_\_\_.  

```

main()
{ int a[6],i;
  for(i=1;i<6;i++)
  { a[i]=9*(i-2+4*(i>3))%5;
    printf("%2d",a[i]); } }

```

 A)-4 0 4 0 4    B)-4 0 4 0 3  
 C)-4 0 4 4 3    D)-4 0 4 4 0 C
- 378 下面是对s的初始化,其中不正确的是\_\_\_\_\_.  
 A) char s[5]="abc"    B) char s[5]='a','b','c';  
 C) char s[5]=""    D) char s[5]="abcdef"; D
- 379 已知字符0的ASCII码为十六进制的30,下面程序的输出是    A) 6    B) 9    C) 0    D) 3  

```

main()
{ union { unsigned char c;
  unsigned int i[4];
} z;
z.i[0]=0x39;
z.i[1]=0x36;
printf("%c\n",z.c);}

```

 B

- 380 以下程序的输出是\_\_\_\_\_ A) 10 B) 11 C) 51 D) 60  

```

struct st
{ int x; int *y;} *p;
int dt[4]={ 10,20,30,40 };
struct st aa[4]={ 50,&dt[0],60,&dt[0],60,&dt[0],60,&dt[0],};
main()
{ p=aa;
printf("%d\n",++(p->x));
}

```

C
- 381 以下程序的输出结果是 A) 12 B) 23 C) 14 D) 32  

```

struct HAR
{ int x, y; struct HAR *p;} h[2];
main()
{ h[0].x=1;h[0].y=2;
h[1].x=3;h[1].y=4;
h[0].p=h[1].p=h;
printf("%d %d \n", (h[0].p)->x, (h[1].p)->y);
}

```

A
- 382 以下程序的输出结果是 A) 4 B) 5 C) 6 D) 0  

```

union myun
{ struct
{ int x, y, z; } u;
int k;
} a;
main()
{ a.u.x=4; a.u.y=5; a.u.z=6;
a.k=0;
printf("%d\n",a.u.x);
}

```

D
- 383 下面程序的输出结果是\_\_\_\_\_ A) 3839 B) 3938 C) 380039 D) 390038  

```

main()
{union
{int k;
char i[2];
}*s,a;
s=&a;
s->i[0]=0x39;s->i[1]=0x38;
printf("%x\n",s->k);
}

```

A
- 384 有如下定义  

```

struct person{char name[9]; int age;};
strict person class[10]={ "Johu", 17, "Paul", 19, "Mary", 18, "Adam", 16};

```

根据上述定义，能输出字母M的语句是  
A) printf("%c\n",class[3].mane);  
B) pfintf("%c\n",class[3].name[1]);  
C) printf("%c\n",class[2].name[1]);  
D) printf("%^c\n",class[2].name[0]);  
D
- 385 有如下函数调用语句  

```

func(rec1,rec2+rec3,(rec4,rec5);

```

该函数调用语句中，含有的实参个数是  
A) 3 B) 4 C) 5 D) 有语法错  
D
- 386 以下选项中，能定义s为合法的结构体变量的是  
A) typedef struct abc B) struct  

```

{ double a; { double a;
char b[10]; char b[10];
} s; } s;
C) struct ABC D) typedef ABC
{ double a; { double a;
char b[10]; char b[10];
} }
ABC s; ABC s;

```

B

- 387 以下对结构体类型变量的定义中, 不正确的是
- A) typedef struct aa { int n; float m; }AA;
- B) #define A struct aa {int n; float m; }td1; AA td1;
- C) struct { int n; float m; }aa;
- D) struct { int n; float m; }td1;
- 388 以下程序运行后, 输出结果是
- ```

fut (int **s, int p[2][3])
{ **s=p[1][1];}
main()
{ int a[2][3]={1, 3, 5, 7, 9, 11}, *p;
p=(int *)malloc(sizeof(int));
fut (&p, a);
printf("%d\n", *p);
}

```
- A) 1      B) 7      C) 9      D) 11
- 389 有语句char str1[10], str2[10]={ "china"}; 则能将字符串china赋给数组str1的正确语句是\_\_\_\_\_。
- A、 str1={ "china"};    B) strcpy (str1, str2);    C) str1=str2;    D) strcpy (str2, str1);
- 390 下列运算符中优先级最高的是\_\_\_\_\_。
- A) <    B) +    C) &&    D) !=
- 391 能正确表示“当x 的取值在[1, 10]和[200, 210]范围内为真, 否则为假”的表达式是\_\_\_\_\_。
- A) (x>=1) && (x<=10) && (x>=200) && (x<=210)
- B) (x>=1) || (x<=10) || (x>=200) || (x<=210)
- C) (x>=1) && (x<=10) || (x>=200) && (x<=210)
- D) (x>=1) || (x<=10) && (x>=200) || (x<=210)
- 392 以下程序的输出结果是
- ```

#include "stdio.h"
struct stu
{ int num;
char name[10];
int age;
};
void fun(struct stu *p)
{ printf("%s\n", (*p).name); }
main()
{
struct stu students[3]={ {9801, "Zhang", 20},
{9802, "Wang", 19},
{9803, "Zhao", 18} };
fun(students+2);
}

```
- A) Zhang    B) Zhao    C) Wang    D) 18
- 393 下面程序的功能是计算正整数2345的各位数字平方和, 请选择填空。
- ```

main()
{ int n, sum=0, n=2345;
do { sum=sum+(n%10)*(n%10);
n= 【2】 ;
}while(n);
printf("sum=%d", sum);}

```
- 【2】** A) n/1000    B) n/100    C) n/10    D) n%10
- 394 下面程序段是输出两个字符串中对应字符相等的字符。请选择填空。
- ```

char x[]="programming";
char y[]="Fortran";
int i=0;
while(x[i]!='\0' && y[i]!='\0')
{ if(x[i]==y[i]) printf("%c", _____);
else i++; }

```
- A) x[i++]    B) y[++i]    C) x[i]    D) y[i]
- 395 以下不正确的定义语句是\_\_\_\_\_。
- A) double x[5]={2.0, 4.0, 6.0, 8.0, 10.0};
- B) int y[5]={0, 1, 3, 5, 7, 9};
- C) char c1[]={' 1' , ' 2' , ' 3' , ' 4' , ' 5' };
- D) char c2[]={' \x10' , ' \xa' , ' \x8' };

- 396 在C语言中, 如果下面的变量都是int类型, 则输出的结果是( )。  

```
sum=5, p=5; p=sum++, p++, ++p;
printf("%d\n", p);
```

A、 7      B、 6      C、 5      D、 4      A
- 397 在scanf( )中, 表示本输入项在读入后不赋给相应变量的格式说明字符是\_\_\_\_  
A) L      B) h      C) C      D) \*      D
- 398 在TC语言中int和short两种数据所占用的内存\_\_\_\_  
A) 均为2个字节      B) 由用户自己定义      C) 由所用机器的机器字长决定      D) 是任意的      A
- 399 在以下定义中, 标识符p\_\_\_\_\_  

```
int (*p)[3]
```

A) 定义不合法      B) 是一个指针数组名, 每个元素是一个指向整数变量的指针  
C) 是一个指针, 它指向一个具有三个元素的一维数组      D) 是一个指向整型变量的指针      C
- 400 一个C程序的执行是从\_\_\_\_\_。  
A) 本程序的main函数开始, 到main函数结束  
B) 本程序文件的第一个函数开始, 到本程序文件的最后一个函数结束  
C) 本程序的main函数开始, 到本程序文件的最后一个函数结束  
D) 本程序文件的第一个函数开始, 到本程序main函数结束      A
- 401 以下叙述正确的是:  
A) 在C程序中, main函数必须位于程序的最前面  
B) C程序的每行中只能写一条语句  
C) C语言本身没有输入输出语句  
D) 在对一个C程序进行编译的过程中, 可发现注释中的拼写错误      C
- 402 以下叙述不正确的是。  
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程序的基本组成单位是函数  
D) 在C程序中, 注释说明只能位于一条语句的后面      D
- 403 C语言规定: 在一个源程序中, main函数的位置 。  
A) 必须在最开始  
B) 必须在系统调用的库函数的后面  
C) 可以任意  
D) 必须在最后      C
- 404 1个C语言程序是由  
A) 1个主程序和若干子程序组成      B) 函数组成  
C) 若干过程组成      D) 若干子程序组成      B
- 405 若x, i, j和k都是int型变量, 则执行下面表达式后x的值为  

```
x=(i=4, j=16, k=32)
```

A) 4      B) 16      C) 32      D) 52      C
- 406 下列四组选项中, 均不是C语言关键字的选项是  
A) define      B) getc      C) include      D) while  
IF      char      scanf      go  
type      printf      case      pow      A
- 407 下面四个选项中, 均是C语言关键字的选项是  
A) auto      B) switch      C) signed      D) if  
enum      typedef      union      struct  
include      continue      scanf      type      B
- 408 下面四个选项中, 均是不合法的用户标识符的选项是。  
A) A      B) float      C) b-a      D) -123  
P\_0      la0      goto      temp  
do      \_A      int      INT      C
- 409 C语言中的标识符只能由字母、数字和下划线三种字符组成, 且第一个字符  
A) 必须为字母      B) 必须为下划线  
C) 必须为字母或下划线      D) 可以是字母, 数字和下划线中任一种字符      C
- 410 下面四个选项中, 均是合法整型常量的选项是:  
A) 160      B) -0xcdf      C) -01      D) -0x48a  
-0xffff      01a      986, 012      2e5  
011      0xe      0668      0x      A
- 411 下面四个选项中, 均是不合法的整型常量的选项是:  
A) -0f1      B) -0Xcdf      C) -018      D) -0x48eg  
-0xffff      017      999      -068  
0011      12, 456      5e2      03f      D





- 430 若有说明语句:char c='\072' 则变量c  
A) 包含1个字符 B) 包含2个字符 A  
C) 包含3个字符 D) 说明不合法, c的值不确定
- 431 若有定义:int a=7; float x=2.5, y=4.7; 则表达式x+a%3\*(int)(x+y)%2/4的值是  
A) 2.500000 B) 2.750000 C) 3.500000 D) 0.000000 A
- 432 sizeof(float)是:  
A) 1个双精度型表达式 B) 1个整型表达式 B  
C) 1种函数调用 D) 1个不合法的表达式
- 433 设变量a是整型, f是实型, i是双精度型, 则表达式10+'a'+i\*f值的数据类型为  
A) int B) float C) double D) 不确定 C
- 434 下面四个选项中, 均是非法常量的选项是。  
A) 'as' B) '\\ ' c) -0x18 D) 0xabc A  
-0fff '01' 01177 '0'  
'\0xa' 12,456 0xf "a"
- 435 在C语言中, char型数据在内存中的存储形式是:  
A) 补码 B) 反码 C) 原码 D) ASCII码 D
- 436 设变量n为float类型, m为int类型, 则以下能实现将n中的数值保留小数点后两位, 第三位进行四舍五入运算的表达式是:  
A) n=(n\*100+0.5)/100.0 B) m=n\*100+0.5, n= m/100.0 B  
C) n=n\*100+0.5/100.0 D) n=(n/100+0.5)\*100.0
- 437 表达式18/4\*sqrt(4.0)/8值的数据类型为:  
A) int B) float C) double D) 不确定 C
- 438 设C语言中, 一个int型数据在内存中占2个字节, 则unsigned int型数据的取值范围为:  
A) 0~255 B) 0~32767 C) 0~65535 D) 0~2147483647 C
- 439 设有说明: char w; int x; float y; double z; 则表达式w\*x+z-y值的数据类型为:  
A) float B) char C) int D) double D
- 440 若w=1, x=2, y=3, z=4, 则条件表达式 w<x?w:y<z?y:z 的值是:  
A) 4 B) 3 C) 2 D) 1 D
- 441 printf函数中用到格式符%5s, 其中数字5表示输出的字符串占用5列。如果字符串长度大于5, 则输出按方式  
A) 从左起输出该字符串, 右补空格 B  
B) 按原字符串长从左向右全部输出  
C) 右对齐输出该字符串, 左补空格
- 442 printf函数中用到格式符%5s, 其中数字5表示输出的字符串占用5列。如果字符串长度小于5, 则输出按方式。  
A) 从左起输出该字符串, 右补空格 C  
B) 按原字符串长从左向右全部输出  
C) 右对齐输出该字符串, 左补空格
- 443 已有定义 int a=-2; 和输出语句: printf("%8lx", a); 以下正确的叙述是:  
A) 整型变量的输出格式符只有%d-种  
B) %x 是格式符的一种, 它可以适用于任何-种类型的数据 D  
C) %x 是格式符的一种, 其变量的值按+六进制输出, 但%8lx 是错误的  
D) %8lx 不是错误的格式符, 其中数字8规定了输出字段的宽度
- 444 若x, y均定义为int型, z定义为double型, 以下不合法的scanf函数调用语句是:  
A) scanf("%d%lx, %le", &x, &y, &z);  
B) scanf("%d \* %d%lf", &x, &y, &z); D  
C) scanf("%x %\* d%o", &x, &y);  
D) scanf("%x%o%6.2f", &x, &y, &z);
- 445 已有如下定义和输入语句, 若要求a1, a2, c1, c2值分别为10, 20, A和B, 当从第1列开始输入数据时, 正确的数据输入方式是  
int a1, a2; char c1, c2;  
scanf("%d%c%d%c", &a1, &c1, &a2, &c2); A  
A) 10A 20B< CR> B) 10 A 20 B<CR>  
C) 10A20<CR> D) 10A20 B<cR>
- 446 已有定义 int x; float y; 且执行scanf("%3d%f", &x, &y); 语句, 若从第1列开始输入数据12345 678(回车), 则x的值为  
A) 12345 B) 123 C) 45 D) 345 B
- 447 已有定义 int x; float y; 且执行scanf("%3d%f", &x, &y); 语句, 若从第1列开始输入数据12345 678(回车), 则y的值:  
A) 无定值 B) 45.0000 C) 678.000000 D) 123.00000 B
- 448 已有如下定义和输入语句, 若要求a1, a2, c1, c2的值分别为10, 20, A, B, 当从第1列开始输入数据时, 正确的数据输入方式是:  
int a1, a2; char c1, c2;  
scanf("%d%d", &a1, &a2); scanf("%c%c", &c1, &c2); D  
A) 1020AB<CR> B) 10 20<CR>AB<CR>  
C) 10 20 AB<CR> D) 10 20AB<CR>

449 已有程序段和输入数据的形式如下, 程序中输入语句的正确形式应当为

```
main()
{   int  a;float f;
    printf("\nInput number:");
    (输入语句)
    printf("\nf=%f,a= %d\n",f,a); }
```

Input number: 4.5 2<CR>

A) scanf("%d,%f",&a,&f); B) scanf("%f,%d",&f,&a)

C) scanf("%d%f",&a,&f); D) scanf("%f%d",&f,&a);

450 根据定义和数据的输入方式, 输入语句的正确形式为:

已有定义: float f1,f2;

数据的输入方式: 4.52<CR>3.5

A) scanf("%f,%f",&f1,&f2); B) scanf("%f%f",&f1,&f2);

C) scanf("%3.2f%2.1f",&f1,&f2); D) scanf("%3.2f%2.1f",&f1,&f2);

451 阅读下程序, 若运行结果为如下形式, 输入输出语句的正确内容是:

```
main()
{int x; float y;printf("enter x,y :")
```

```
(输入语句) (输出语句) }
```

输入形式 enter x,y: 2 3.4

输出形式 x+y=5.40

A) scanf("%d,%f",&x,&y);printf("\nx+y= %4.2f",x+y);

B) scanf("%d%f",&x,&y);printf("\nx+y=%4.2f",x+y);

C) scanf("%d%f",&x,&y); printf("\nx+y=%6.1f",x+y);

D) scanf("%d%3.1f",&x,&y);printf("\nx+y=%4.2f",x+y);

452 以下说法正确的是:

A) 输入项可以为-实型常量, 如scanf("%f",3.5);

B) 只有格式控制, 没有输入项, 也能进行正确输入, 如scanf("a=%d,b=%d");

C) 当输入一个实型数据时, 格式控制部分应规定小数点后的位数, 如scanf("%4.2f",&f);

D) 当输入数据时, 必须指明变量的地址, 如scanf("%f",&f);

453 根据下面的程序及数据的输入方式和输出形式, 程序中输入语句的正确形式应该为:

```
main()
{char ch1 ,ch2 ,ch3; (输入语句)
```

```
printf("%c%c%c",ch1,ch2,ch3);}
```

输入形式: A B C

输出形式: A B

A) scanf("%c%c%c",&ch1,&ch2,&ch3);

B) scanf("%c,%c,%c",&ch1,&ch2,&ch3);

C) scanf("%c %c %c",&ch1,&ch2,&ch3);

D) scanf("%c%c",&ch1,&ch2,&ch3);

454 有输入语句: scanf("a=%d,b=%d,c=%d",&a,&b,&c);为使变量a的值为1,b为3,c为2,从键盘输入数据的正确形式应当是

A) 132(回车)

B) 1,3,2(回车)

C) a=1b=3c=2(回车)

D) a=1,b=3,c=2(回车)

455 以下能正确地定义整型变量a,b和c 并为其赋初值5的语句是

A) int a=b=c=5; B) int a,b, c= 5;

C) a=5,b=5,c=5; D) a= b= c=5;

456 已知ch是字符型变量, 下面不正确的赋值语句是

A) ch='a+b'; B) ch='\0'; C) ch='7'+ '9'; D) ch=5+9;

457 已知ch是字符型变量, 下面正确的赋值语句是

A) ch= '123'; B) ch= '\xff'; C) ch= '\08'; D) ch="\\";

458 若有以下定义, 则正确的赋值语句是:

```
int a,b=5;float x;
```

A) a=1,b=2, B) b++; C) a=b=5 D) b=int(x);

459 设x,y均为float型变量, 则以下不合法的赋值语句是

A) ++x; B) y=(x%2)/10; C) x\*=y+8; D) x=y=0;

460 x,y,z均为int型变量, 则执行语句x=(y=(z=10)+5)-5;后, x,y和c的值是:

A) x=10 B) x=10 C) x=10 D) x=10

y=15 y=10 y=10 y=5

z=10 z=10 z=15 z=10

461 逻辑运算符两侧运算对象的数据类型\_\_\_\_\_。

A) 只能是0或1

B) 只能是0或非0正数

C) 只能是整型或字符型数据

D) 可以是任何类型的数据

- 462 下列关于运算符优先顺序的描述中正确的是\_\_\_\_\_。
- A) 关系运算符< 算术运算符< 赋值运算符< 逻辑与运算符  
 B) 逻辑运算符< 关系运算符< 算术运算符< 赋值运算符  
 C) 赋值运算符< 逻辑与运算符< 关系运算符< 算术运算符  
 D) 算术运算符< 关系运算符< 赋值运算符< 逻辑与运算符
- 463 表示图中坐标轴上阴影部分的正确表达式是：
- ////////|////////|  
 -----> X  
 a          b          c
- A) (x<=a) && (x>=b) && (x<=c)  
 B) (x<=a) || (b<=x<=c)  
 C) (x<=a) || (x>=b) && (x<=c)  
 D) (x<=a) && (b<=x<=c)
- 464 判断char型变量ch是否为大写字母的正确表达式是\_\_\_\_\_。
- A) 'A' <=ch<='Z'                      B) (ch>='A') & (ch<='Z')  
 C) (ch>='A') && (ch<='Z')    D) ('A' <=ch) AND ('Z' >=ch)
- 465 设x, y 和z是int型变量, 且x = 3, y= 4, z= 5, 则下面表达式中值为0的是
- A) 'y' && 'y'  
 B) x<y  
 C) x||y+z&&y-z  
 D) !(x<y) && !z||1)
- 466 已知x =43, ch='A', y=0则表达式(x >= y && ch<'B' && !Y) 的值是
- A) 0    B) 语法错    C) 1    D) "假"
- 467 若希望当A的值为奇数时, 表达式的值为"真", A的值为偶数 表达式的值为"假"。则以下不能满足要求的表达式是:
- A) A%2==1          B) !(A%2==0)          C) !(A%2)          D) A%2
- 468 以下程序的运行结果是:
- ```
#include "stdio.h"
main()
{ int a,b,d=241; a=d/100%9; b=(-1)&&(-1);
  printf("%d,%d",a,b); }
```
- A) 6, 1    B) 2, 1    C) 6, 0    D) 2, 0
- 469 执行以下语句后a的值为:
- ```
int a,b,c; a=b=c=1; ++a||++b&&++c;
```
- A) 错误    B) 0    C) 2    D) 1
- 470 以下if语句语法正确的是\_\_\_\_\_。
- A) if(x>0)    printf("%f", x)  
      else    printf("%f", --x);  
 B) if(x>0) { x=x+y;    printf("%f", x); }  
      else    printf("%f", -x);  
 C) if(x>0) { x=x+y;    prinrf("%f", x); };  
      else    printf("%f", -x);  
 D) if(x>0) { x=x+y;    printf("%f", x) }  
      else    printf("%f", -x);
- 471 以下程序的运行结果是 :
- ```
main()
{ int m=5;
  if(m++>5)    printf(" %d\n", m);
  else        printf("%d\n", m--); }
```
- A) 4    B) 5    C) 6    D) 7
- 472 当a=1, b=3, c=5, d=4 , 执行完下面1段程序后x的值是 :
- ```
if(a<b)
if(c<d) x=1;
else
if(a<c)
if(b<d) x=2;
else x= 3;
else x=6;
else x=7;
```
- A) 18    B) 2    C) 3    D) 6

- 473 以下程序的输出结果是：
- ```
main()
{int  a=100,x=10,y=20,ok1=5,ok2=0;
  if(x<y)
  if(y!=10)
  if(!ok1)  a=1;
  else
  if(ok2)  a=10;
  a=-1;
  printf("%d\n",a);
}
```
- A) 1      B) 0      C) -1      D) 值不确定 C
- 474 以下程序的输出结果是：
- ```
main()
{ int  x=2,y=-1,z=2;
  if (x<y)
  if(y<0)  z= 0;
  else    z+=1;
  printf("%d\n",z);
}
```
- A) 3      B) 2      C) 1      D) 0 B
- 475 以下不正确的语句为：
- A) if(x>y);  
 B) if(x=y) && (x!=0) x+=y;  
 C) if(x!=y) scanf("%d",&x); else scanf("%d",&y);  
 D) if(x<y) { x++ ;y++; }
- A) 3      B) 2      C) 1      D) 0 B
- 476 若有条件表达式(exp) ? a++:b--,则以下表达式中能完全等价于表达式(exp)的是：
- A) (exp==0)      B) (exp!=0)  
 C) (exp==1)      D) (exp!=1) B
- 477 若运行时给变量x 输入12,则以下程序的运行结果是：
- ```
main()
{ int  x,y;    scanf("%d",&x);
  y=x>12 ? x+10 : x-12 ;
  printf("%d\n",y); }
```
- A) 0      B) 22      C) 12      D) 10 A
- 478 以下程序的运行结果是：
- ```
main()
{ int k=4,a=3,b=2,c=1;
  printf("\n%d\n",k<a ? k:c<b ? c :a); }
```
- A) 4      B) 3      C) 2      D) 1 D
- 479 执行以下程序段后、变量a,b,c的值分别是\_\_\_\_\_。
- ```
int  x=10,y=9;    int  a,b,c;
a = (--x == y++) ? --x : ++y ;
b = x++;
c = y;
```
- A) a=9,b= 9,c= 9      B) a=8,b=8,c=10  
 C) a=9,b= 10,c=9      D) a=1,b= 11,c=10 B
- 480 执行以下程序段后的输出结果是
- ```
int  w=3,  z=7,x =10;
printf("%d\n",x>10 ? x+100 : x-10 );
printf("%d\n",w++|z++ );
printf("%d\n",!w>z);
printf("%d\n",w&&z);
```
- A) 0 0 0 0      B) 0 1 1 0      C) 0 1 0 1      D) 1 0 1 0 C
- 481 有以下程序段
- ```
int  x=0,s=0;
while (!x!=0)  s+=++x;
printf("%d",s);
```
- A) 运行程序段后输出0      B) 运行程序段后输出1  
 C) 程序段中的控制表达式是非法的      D) 程序段执行无限次 B
- 482 以下正确的描述是。
- A) continue语句的作用是结束整个循环的执行  
 B) 只能在循环体内和switch语句体内使用break语句  
 C) 在循环体内使用break语句或continue语句的作用相同  
 D) 从多层循环嵌套中退出时, 只能使用goto语句 B

483 下面程序的功能是将从小到大的1对数，由小到大排序输出。当输入1对相等数时结束循环，请选择填空：

```
main()
{ int a,b,t;
  scanf("%d%d",&a,&b);
  while( 【1】 )
  { if(a>b) { t=a;a=b;b=t;}
    printf("%d,%d",a,b);
    scanf("%d%d",&a,&b);} }
```

【1】 A) !a=b B) a!=b C) a==b D) a=b

484 程序的功能是从键盘输入的字符串中统计出大写字母和小写字母的个数，并输出m、n中的较大者，请选择填空：

```
#include "stdio.h"
main()
{int m=0,n=0;
 char c;
 while(( 【1】 )!='\n')
 { if(c>='A' && C<='Z') m++ ;
   if(c>='a' && c<='z') n++; }
 printf("%d\n", m<n? n:m); }
```

485 下面程序的功能是将小写字母变成对应大写字母后的第二个字母。其中y变成A，z变成B。请选择填空。

```
main()
{ char c;
  while((c=getchar())!='\n')
  { if(c>='a' && c<='z') c-=30;
    if(c>'z' && c<='z'+2)
      【2】; }
  printf("%c",c) }
```

【2】 A) c='B' B) c='A' C) c-=26 D) c=c+26

486 下面程序的功能是在输入的一批正整数中求出最大者，输入0结束循环，请选择填空。

```
main()
{ int a,max= 0;
  scanf("%d",&a)
  while( 【1】 )
  { if(max<a max=a;
    scanf("%d",&a); }
  printf("%d",max); }
```

【1】 A) a==0 B) a C) !a==1 D) !a

487 下面程序的运行结果是。

```
main()
{ int num= 0;
  while(num<=2)
  { num++;
    printf ("%d\n", num); } }
```

A) 1 B) 1 c) 1 D) 1  
2 2 2  
3 3  
4

488 以下程序段

```
x=-1;
do
{ x=x*x; }while(!x);
```

A) 是死循环 B) 循环执行二次 C) 循环执行1次 D) 有语法错误

489 下面程序是从键盘输入学号，然后输出学号中百位数字是3的学号，输入0时结束循环。请选择填空。

```
main()
{ long int num;
  scanf("%ld",&num);
  do { if( 【1】 ) printf("%ld",num);
    scanf("%ld",&num);
  }while(!num==0);}
```

【1】 A) num>300 B) num/100%10==3 C) !num==3 D) !num!=3

490 下面程序的功能是把316表示为两个加数的和，使两个加数分别能被13和11整除。请选择填空。

```
main()
{ int i=0,j,k;
  do{ i++;k=316-13*i; }while( 【1】 ); j=k/11;
  printf(" 316=13* %d+11*%d",i,j); }
```

【1】 A) k/11 B) k%11 C) k/11==0 D) k/11== 0

491 下面程序的运行结果是:

```
main()
{ int y=10;
  do{y--;}while(--y);
  printf("%d\n",y--);}
A)-1 B) 1 C) 8 D) 0
```

D

492 若运行以下程序时,从键盘输入ADescriptor<CR>(CR表示回车),则下面程序的运行结果是:

```
main()
{ char c;
  int v0=0,v1=0,v2=0;
  do{
    switch(c=getchar())
    { case 'a':      case 'A' :
      case 'e':      case 'E' :
      case 'i':      case 'I' :
      case 'o':      case 'O' :
      case 'u':      case 'U' : v1+=1;
      default: v0+=1; v2+=1; }
    }while(c!='\n');
  printf("v0=%d,v1=%d,v2=%d\n",v0,v1,v2); }
```

D

- A) v0=7, v1=4, v2=7 B) v0=8, v1=4, v2=8  
C) v0=11, v1=4, v2=11 D) v0=12, v1=4, v2=12

493 下面程序的运行结果是:

```
main()
{ int a=1,b=10;
  do
  { b-=a;a++;}while(b--<0);
  printf("a=%d,b=%d\n",a,b); }
```

B

- A) a=3, b=11 B) a=2, b=8 C) a=1, b=-1 D) a=4, b=9

494 下面有关for循环的正确描述是:

- A) for循环只能用于循环次数已经确定的情况  
B) for循环是先执行循环体语句,后判断表达式  
C) 在for循环中,不能用break语句跳出循环体  
D) for循环的循环体语句中,可以包含多条语句,但必须用花括号括起来

D

495 对for(表达式1;;表达式3)可理解为:

- A) for(表达式1;0;表达式3)  
B) for(表达式1;1;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)

B

496 若i为整型变量,则以下循环执行次数是:

for(i=2;i==0;) printf("%d",i-- );

- A) 无限次 B) 0次 C) 1次 D) 2次

B

497 以下for循环的执行次数是:

for(x=0,y=0;(y=123)&&(x<4);x++);

- A) 是无限循环 B) 循环次数不定 C) 执行4次 D) 执行3次

C

498 以下不是无限循环的语句为:

- A) for(y=0,x=1;x>++y;x=i++) i=x ;  
B) for(;; x+=i);  
C) while(1) { x++;}  
D) for(i=10; ;i--) sum+=i;

A

499 下面程序段的运行结果是:

```
for(y=1;y<10;) y=((x=3*y,x+1),x-1);
printf("x=%d,y=%d",x,y);
```

- A) x=27,y=27 B) x=12,y=13 C) x=15,y=14 D) x=y=27

C

500 下面程序段的运行结果是

```
for(x=3;x<6;x++) printf((x %2)?("**%d"):( " ##%d\n"),x);
```

- A)\*\*3 ##4 B) ##3 ##4 C) ##3 \*\*4 D)\*\*3 \*\*4  
\*\*5 \*\*5 \*\*5 ##5

A

501 以下各组选项中,均能正确定义二维实型数组a的选项是

- A) float a[3][4];  
float a[][4];  
float a[3][]={{1},{0}};  
C) float a[3][4];  
static float a[][4]={{0},{0}};  
auto float a[][4]={{0},{0},{0}};
- B) float a(3,4);  
float a[3][4];  
float a[][]={{0},{0}};  
D) float a[3][4];  
float a[3][];  
float a[][4];

C

502 下面程序的功能是计算:至50中是7的倍数的数值之和, 请选择填空。

```
main()  
{ int i,sum= 0;  
  for(i=1;i<=50;i++)  
    if([1]) sum+=i;  
  printf("%d",sum); }  
A) (int) (i/7)==i/7  
B) (int) i/7==i/7  
C) i%7=0  
D) i%7==0
```

D

503 下面程序的功能是计算 1至10之间的奇数之和及偶数之和。请选择填空;

```
main()  
{ int a,b,c,i;  
  a=c=0;  
  for(i=0;i<=10;i+=2)  
  { a+=i;  
    [1];  
    c+=b; }  
  printf("偶数之和=%d\n",a);  
  printf("奇数之和=%d\n",c-11); }  
[1] A )b=i-- B) b=i+1 C) b=i++ D)b=i-1
```

B

504 下面程序的运行结果是:

```
main()  
{ int i;  
  for(i=1;i<=5;i++)  
  switch(i%5)  
  { case 0 : printf("*");break;  
    case 1 : printf("#");break;  
    default : printf("\n");  
    case 2 : printf("&");  
  } }  
A) #& B) #& C) # D) #&  
&  
& &*
```

B

505 下面程序的运行结果是:

```
main()  
{ int x,i,j;  
  for(i=1;i<=100;i++)  
  { x=i;  
    if(++x%2==0)  
      if(++x%3==0)  
        if(++x%7==0)  
          printf("%d ",x);  
  } }
```

D

A) 39 81 B) 42 84 C) 26 68 D) 28 70

506 下面程序段的功能是计算1000!的末尾含有多少个零。请选择填空。(提示:只要算出1000!中含有因数5的个数即可)

```
for(k=0,i=5;m<=1000; m+=5)  
  while(【1】) { k++; m=m/5; }  
【1】 A)m%5==0 B)m=m%5==0 C)m%5==0 D)m%5!=0
```

C

507 下面程序的运行结果是:

```
main()  
{ int i,b,k=0;  
  for(i=1;i<=5;i++)  
  { b=i%2;  
    while(b-->=0) k++; }  
  printf("%d,%d",k,b); }  
A) 3,-1 B) 8,-1 C) 3,0 D) 8,-2
```

D

508 下面程序段:

```
for(t=1;t<=100;t++)  
{ scanf("%d",&x);  
  if(x<0) continue;  
  printf("%3d",t); }  
A) 当x<0时整个循环结束  
B) x>=0时什么也不输出  
C) printf函数永远也不执行  
D) 最多允许输出100个非负整数
```

D



- 509 下面程序段：
- ```

x=3;
do
{
y=x--;
if(!y) { printf("x");continue; }
printf("#");
}while(1<=x<=2);

```
- A) 将输出##      B) 将输出##\*  
C) 是死循环      D) 含有不合法的控制表达式
- 510 以下描述正确的是
- A) goto语句只能用于退出多层循环  
B) switch语句中不能出现continue语句  
C) 只能用continue语句来终止本次循环  
D) 在循环中break语句不能独立出现
- 511 与下面程序段等价的是：
- ```

for(n=100;n<= 200; n++)
{
if (n%3==0) continue;
printf("%4d",n); }

```
- A) for(n=100;(n%3)&& n<=200;n++)      printf("%4d",n);  
B) for(n=100;(n%3)|| n<=200;n++)      printf("%4d",n);  
C) for(n=100;n<=200;n++) if(n%3!=0)      printf("%4d",n)  
D) for(n=100;n<=200; n++)  
    { if(n%3)      printf("%4d",n);  
      else      continue;  
      break;      }
- 512 下面程序的运行结果是：
- ```

main()
{
int k=0;
char c='A';
do
{
switch(c++)
{
case 'A': k++;break;
case 'B': k--;
case 'C': k+=2;break;
case 'D': k=k%2;continue;
case 'E': k=k*10;break;
default:k=k/3; }
k++;
}while(c<'G');
printf("k=%d",k); }

```
- A) k=3      B) k=4      C) k=2      D) k=0
- 513 若运行以下程序时，从键盘输入3.6,2.4<CR>，(<CR>表示回车)，则下面程序的运行结果是：
- ```

#include <stdio.h>      #include <math.h>
main()
{
float x,y,z;
scanf("%f,%f",&x,&y);
z=x/y;
while(1)
{
if(fabs(z)>1.0) { x=y;y=z;z=x/y; }
else break; }
printf("%f",y); }

```
- A) 1.5      B) 1.6      C) 2.0      D) 2.4
- 514 下面程序的运行结果是：
- ```

main()
{
int a,b;
for(a=1,b=1;a<=100;a++)
{
if (b>=20) break;
if(b%3==1) {b+=3;continue;}
b-=5; }
printf("%d\n",a); }

```
- A) 7      B) 8      C) 9      D) 10

- 515 下面程序的运行结果是：
- ```
main()
{ int i;
  for(i=1;i<=5;i++)
  {   if (i%2) printf("*");
      else continue;
      printf("#"); }
  printf("$"); }
```
- A) \*##\*\$    B) \*\*##\*\$    C) \*\*\*#\$    D) \*\*\*\*\$ B
- 516 下面程序的运行结果是：
- ```
main()
{ int i, j, a=0;
  for(i=0;i<2;i++)
  {   for (j=0;j<=4;j++)
      {   if (j%2) break;
          a++; }
      a++; }
  printf("%d\n",a); }
```
- A) 4    B) 5    C) 6    D) 7 A
- 517 以下对1维整型数组a的正确说明是\_\_\_\_\_。
- A) float a(10);    B) int n= 10, a[n]; D  
 C) int n; int a[n];    D) #define SIZE 10; int a[SIZE];
- 518 若有说明:int a[10];则对a数组元素的正确引用是\_\_\_\_\_。
- A) a[10]    B) a[3.5]    C) a[5]    D) a[10\*10] C
- 519 以下能对1维数组a进行正确初始化的语句是\_\_\_\_\_。
- A) int a[10]=(0,0,0,0,0)    B) int a[10]={} C  
 C) int a[]={0};    D) int a[10]={10\*1};
- 520 以下对二维数组a的正确说明是\_\_\_\_\_。
- A) int a[3][];    B) float a(3,4); C  
 C) double a[1][4];    D) float a(3)(4);
- 521 若有说明:int a[3][4]; 则对a数组元素的正确引用是\_\_\_\_\_。
- A) a[2][4]    B) a[1,3]    C) a[1+1][0]    D) a(2)(1); C
- 522 若有说明:int a[3][4];则对a数组元素的非法引用是\_\_\_\_\_。
- A) a[0][2\*1]    B) a[1][3]    C) a[4-2][0]    D) a[0][4] D
- 523 以下能对二维数组a进行正确初始化的语句是\_\_\_\_\_。
- A) int a[2][]={{1,0,1},{5,2,3}}; B  
 B) int a[][3]={{1,2,3},{4,5,6}};  
 C) int a[2][4]={{1,2,3},{4,5},{6}};  
 D) int a[][]={{1,0,1},{},{1,1}};
- 524 若有说明: int a[3][4]={0};则下面正确的叙述是
- A) 只有元素a[0][0]可得到初值0  
 B) 此说明语句不正确:。  
 C) 数组a中各元素都可得到初值,但其值不一定为0。  
 D) 数组a中每个元素均可得到初值0 D
- 525 若有说明:int a[][4]={0,0};则下面不正确的叙述是\_\_\_\_\_。
- A) 数组a的每个元素都可得到初值0  
 B) 二维数组a的第1维大小为1 D  
 C) 因为二维数组a中第二维大小的值除以初值个数的商为1,故数组a行数为1  
 D) 只有元素a[0][0]和a[0][1]可得初值0,其余元素均得不到初值0
- 526 若有说明:int a[3][4];则数组a各元素
- A) 可在程序的运行阶段得到初值0  
 B) 可在程序的编译阶段得到初值0 B  
 C) 不能得到确定的初值  
 D) 可在程序的编译或运行阶段得初值0
- 527 下面程序(每行程序前面的数字表示行号)
- ```
1 main()
2 {
3   int a[3]={3,0};
4   int i;
5   for(i=0;i<3;i++) scanf("%d",&a[i]);
6   for(i=1;i<3;i++) a[0]=a[0]+a[i];
7   printf("%d\n",a[0]); }
```
- A) 没有错误    B) 第3行有错误 A  
 C) 第5行有错误    D) 第7行没有错误

528 下面程序\_\_\_\_\_ (每行程序前面的数字表示行号)。

```
1  main()  
2  {  
3      float a[10]={0,0};  
4      int i;  
5      for(i=0;i<10;i++) scanf("%d",&a[i]);  
6      for(i=0;i<10;i++) a[0]=a[0]+a[i];  
7      printf("%d\n",a[0]);  
8  }
```

- A) 没有错误                      B) 第3行有错误  
C) 第5行有错误                  D) 第7行有错误

529 下面程序有错的行是

```
1  main()  
2  {  
3      int a[3]={1};  
4      int i;  
5      scanf("%d",&a);  
6      for(i=1;i<3;i++) a[0]=a[0]+a[i];  
7      printf("a[0]=%d\n",a[0]);  
8  }
```

- A) 3                      B) 6                      C) 7                      D) 5

530 下面程序 (每行程序前面的数字表示行号)

```
1  main()  
2  {  
3      int a[3]={0};  
4      int i;  
5      for(i=0;i<3;i++) scanf("%d",&a[i]);  
6      for(i=1;i<4;i++) a[0]=a[0]+a[i];  
7      printf("%d\n",a[0]);  
8  }
```

- A) 没有错误                      B) 第3行有错误  
C) 第5行有错误                  D) 第6行有错误

531 若二维数组a有m列,则计算任1元素a[i][j]在数组中位置的公式为: (假设a[0][0]位于数组的第一个位置上)

- A)  $i*m+j$                       B)  $j*p+1$                       C)  $i*m+j-1$                       D)  $i*m+j+1$

532 对以下说明语句的正确理解是

```
int a[10]={6,7,8,9,10};
```

- A) 将5个初值依次赋给a[1]至a[5]  
B) 将5个初值依次赋给a[0]至a[4]  
C) 将5个初值依次赋给a[6]至a[10]  
D) 因为数组长度与初值的个数不相同,所以此语句不正确

533 若二维数组a有m列,则在a[i][j]前的元素个数为\_\_\_\_\_.

- A)  $j*m+j$                       B)  $i*m+j$                       C)  $i*m+j$                       D)  $i*m+j+1$

534 若有以下程序段:

```
int a[]={4,0,2,3,1}, i, j, t;  
for(i=1;i<5;i++)  
{ t=a[i]; j=i-1;  
  while(j>=0&& t>a[j])  
  { a[j+1]=a[j]; j--; }  
  .....  
}
```

则该程序段的功能是 \_\_\_\_\_.

- A) 对数组a进行插入排序(升序)  
B) 对数组a进行插入排序(降序)  
C) 对数组a进行选择排序(升序)  
D) 对数组a进行选择排序(降序)

- 535 下面程序的运行结果是\_\_\_\_。
- ```
main()
{ int a[6][6], i, j;
  for(i=1; i<6; i++)
    for(j=1; j<6; j++)
      a[i][j]=(i/j)*(j/i);
  for(i=1; i<6; i++)
    { for(j=1; j<6; j++)
      printf("%2d", a[i][j]);
      printf("\n"); } }
```
- A) 11111      B) 00001      C) 10000      D) 10001  
 11111      00010      01000      01010  
 11111      00100      00100      00100  
 11111      01000      00010      01010  
 11111      10000      00001      10001
- 536 下面程序段的运行结果是\_\_\_\_。
- ```
char c[5]={'a','b','\0','c','\0'};
printf("%s", c);
```
- A) a' b'      B) ab      C) ab c      D) ab
- 537 对两个数组a和b进行如下初始化
- ```
char a[]="ABCDEF";
char b[]={ 'A' , 'B' , 'C' , 'D' , 'E' , 'F' };
```
- 则以下叙述正确的是\_\_\_\_\_。
- A) a与b数组完全相同      B) a与b长度相同  
 C) a和b中都存放字符串      D) a数组比b数组长度长
- 538 有两个字符数组a、b, 则以下正确的输入格式是\_\_\_\_\_。
- A) gets(a, b);      B) scanf("%s%s", a, b);  
 C) scanf("%s%s", &a, &b);      D) gets("a"), gets("b");
- 539 下面程序段的运行结果是\_\_\_\_。(其中-表示空格)
- ```
char a[7]="abcdef";
char b[4]="ABC";
strcpy(a, b);
printf("%c", a[5]);
```
- A) -      B) \0      C) e      D) f
- 540 下面程序段的运行结果是\_\_\_\_\_。
- ```
char c[]="\t\t\v\\0will\n";
printf("%d", strlen(c));
```
- A) 14      B) 3      C) 9      D) 字符串中有非法字符, 输出值不确定
- 541 下面描述正确的是\_\_\_\_\_。
- A) 两个字符串所包含的字符个数相同时, 才能比较字符串  
 B) 字符个数多的字符串比字符个数少的字符串大  
 C) 字符串"STOP "与"STOp"相等  
 D) 字符串"17hat"小于字符串"he"
- 542 下述对C语言字符数组的描述中错误的是
- A) 字符数组可以存放字符串  
 B) 字符数组的字符串可以整体输入、输出  
 C) 可以在赋值语句中通过赋值运算符"="对字符数组整体赋值  
 D) 不可以用关系运算符对字符数组中的字符串进行比较
- 543 有已排好序的字符串a, 下面的程序是将字符串s中的每个字符按a中元素的规律插入到a中。请选择填空。
- ```
main()
{ char a[20]="cehiknqtw";
  char s[]="fbla";
  int i, k, j;
  for(k=0; s[k]!='\0'; k++)
  { j=0;
    while(s[k]>a[j]&& a[j]!='\0') j++;
    for(i=strlen(a); i>=j; i--) 【2】;
    a[j]=s[k]; }
  puts(a); }
```
- A) a[i]=a[i+1]      B) a[i+1]=a[i];  
 C) a[i]=a[i-1]      D) a[i-1]=a[i];

544 下面程序的功能是从键盘输入1行字符,统计其中有多少个单词,单词之间用空格分隔。请选择填空。

```
main()
{
    char s[80],c1,c2='';
    int i=0,num=0;
    gets(s);
    while(s[i]!='\0')
    {
        c1=s[i];
        if(i==0) c2='';
        else c2=s[i-1];
        if(【1】) num++;
        i++;
    }
    printf("There are %d words.\n",num);
}
```

【1】 A) c1=='&& c2==' B) c1!='&& c2=='  
C) c1=='&& c2!=' D) c1!='&& c2!='

B

545 下面程序的运行结果是

```
main()
{
    char ch[7]={"12ab56"};
    int i,s=0;
    for(i=0;ch[i]>='0'&&ch[i]<='9';i+=2)
        s=10*s+ch[i]-'0';
    printf("%d\n",s);
}
```

A) 1 B) 1256 C) 12ab56 D) 1  
2  
5  
6

A

546 当运行以下程序时,从键盘输入:aa bb<CR>cc dd<CR> (<CR>表示回车),则下面程序的运行结果是

```
main()
{
    char a1[5],a2[5],a3[5],a4[5];
    scanf("%s%s",a1,a2);
    gets(a3); gets(a4);
    puts(a1); puts(a2);
    puts(a3); puts(a4);
}
```

A) aa bb bb cc cc dd dd ee  
bb bb bb cc dd  
cc dd dd ee

A

547 当运行以下程序时,从键盘输入:ab<CR>c<CR>dd<CR> (<CR>表示回车),则下面程序的运行结果是

```
#include<stdio.h> #define N 6
main()
{
    char c[N];
    int i=0;
    for(;i<N;c[i]=getchar(),i++);
    for(i=0;i<N;i++) putchar(c[i]);
}
```

A) abcdef B) a b c d e f  
b c c  
c d d

D

548 下面程序的运行结果是

```
main()
{
    char str[]="SSSWLIA",c;
    int k;
    for(k=2;(c=str[k])!='\0';k++)
    {
        switch(c)
        {
            case 'I': ++k;break;
            case 'L': continue;
            default : putchar(c);continue;
        }
        putchar('*');
    }
}
```

A) SSW\* B) SW\* C) SW\*A D) SW

B

549 标准函数 fgets(s, n, f) 的功能是

- A) 从文件f中读取长度为n的字符串存入指针s所指的内存  
B) 从文件f中读取长度不超过n-1的字符串存入指针s所指的内存  
C) 从文件f中读取n个字符串存入指针s所指的内存  
D) 从文件f中读取长度为n-1的字符串存入指针s所指的内存

B

- 550 rewind函数的作用是\_\_\_\_\_。  
A、重新打开文件 B、使文件位置指针重新回到文件末 C  
C、使文件位置指针重新回到文件的开始 D、返回文件长度值
- 551 假定建立了以下链表结构，指针p、q分别指向如图所示的结点，则以下可以将q所指结点从链表中删除并释放该结点的语句组是  
A) free(q); p->next=q->next; B) (\*p).next=(\*q).next; free(q); B  
C) q=(\*q).next; (\*p).next=q; free(q); D) q=q->next; p->next=q; p=p->next; free(p);
- 552 假定以下程序经编译和连接后生成可执行文件PROG.EXE，如果在此可执行文件所在目录的DOS提示符下键入:PROG ABCDEFGH IJKL<回车>，则输出结果为  
A) ABCDEFG B) IJHL C) ABCDEFGHIJKL D) IJKLABCDEFGH D  
main( int argc, char \*argv[]))  
{ while(--argc>0) printf("%s",argv[argc]);  
printf("\n");}
- 553 若fp是指向某文件的指针，且已读到文件末尾，则库函数feof(fp)的返回值是\_\_\_\_\_  
A) EOF B) -1 C) 非零值 D) NULL C
- 554 若要打开A盘上user子目录下名为abc.txt的文本文件进行读、写操作，下面符合此要求的函数调用是  
A) fopen("A:\user\abc.txt","r") B) fopen("A:\\user\\abc.txt","r+") B  
C) fopen("A:\user\abc.txt","rb") C) fopen("A:\\user\\abc.txt","w")
- 555 若要用下面的程序片段使指针变量p指向一个存储整型变量的动态存储单元：  
int \*p;  
p=\_\_\_\_\_ malloc( sizeof(int)); D  
则应填入 A)int B)int \* C)(\*int) D)(int \*)
- 556 若以下程序所生成的可执行文件名为file1.exe；当打入以下命令行执行该程序时；  
FILE1 CHINA BEIJING SHANGHAI 程序的输出结果是  
A、 XHIN VWIJINF AHANJHAI B、 FILE1 CHINA BEIJING  
C、 C B S D、 F C B  
main(int arge,char \* argv [])  
{ char ch [7]={"65ab21"};  
int i,s=0;  
for (i=0;ch [i]>='0'&&ch [i]!='9';i+=2)  
s=10\*s+ch[i]-'0';  
print ("%d\n",s); } A
- 557 以数据块为单位对数据进行整体 读写时如果ptr是指向内存中数据块的首地址，fp是文件指针，那么数据块中每个数据项的大小为( )。  
A) sizeof(\*ptr) B) \*ptr C) sizeof(ptr) D) sizeof(\*fp) A
- 558 在C程序中，可把整型数以二进制形式存放到文件中的函数是  
A) fprintf函数 B) fread函数 C) fwrite函数 D) fputc函数 A
- 559 在C语言中，fclose()函数返回\_\_\_\_时，表示关闭不成功  
(A) 0 (B) -1 (C) EOF (D) 非零值 D