|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | t为int类型，进人下面的循环之前，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 |  | |  | |  | |  | |  | |  | |  | |  |
| 2 | 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; |  | |  | |  | |  | |  | |  | |  | |  |
| 3 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 4 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 5 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 6 | 以下程序的输出结果是\_\_\_\_ A、5 B、6 C、23 D、26 i=5;  i=++i\*4+2; |  | |  | |  | |  | |  | |  | |  | |  |
| 7 | 以下程序的输出结果是\_\_\_\_ A、5 B、6 C、23 D、26  i=5;  i=i++\*4+2; |  | |  | |  | |  | |  | |  | |  | |  |
| 8 | 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);  } |  | |  | |  | |  | |  | |  | |  | |  |
| 9 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 10 | 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。 |  | |  | |  | |  | |  | |  | |  | |  |
| 11 | #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 |  | |  | |  | |  | |  | |  | |  | |  |
| 12 | 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; | A |  | |  | |  | |  | |  | |  | |  | |  |
| 13 | 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}; | D |  | |  | |  | |  | |  | |  | |  | |  |
| 14 | 下面程序的功能是将字符串5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]; | A |  | |  | |  | |  | |  | |  | |  | |  |
| 15 | 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]);  } | B |  | |  | |  | |  | |  | |  | |  | |  |
| 16 | 判断字符串s1是否大于字符串s2应当使用\_\_\_\_\_. A) if(sl>s2) B) if(strcmp(s1,s2)) C) if(strcmp(s2,sl)>0) D) if(strcmp(s1,s2)>0) | D |  | |  | |  | |  | |  | |  | |  | |  |
| 17 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 18 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 19 | 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) (x= 6,x+1,y= 6,x+y) D) (y=6,y+l,x = y,x+1) | C |  | |  | |  | |  | |  | |  | |  | |  |
| 20 | 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 | A |  | |  | |  | |  | |  | |  | |  | |  |
| 21 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 22 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 23 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 24 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 25 | 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(strcpy(a,b)) D) if(!strcmp(a,b)) | D |  | |  | |  | |  | |  | |  | |  | |  |
| 26 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 27 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 28 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 29 | 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);  C) scanf("%d%d%d",a,b,d); D) scanf("%d%d%d",a,b,\*d); | A |  | |  | |  | |  | |  | |  | |  | |  |
| 30 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 31 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 32 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 33 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 34 | 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] | B |  | |  | |  | |  | |  | |  | |  | |  |
| 35 | 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][l]  int t[3][3]， \*pt[3]， k;  for (k=0;k<3;k++) pt[k]=&t[k][0]; | C |  | |  | |  | |  | |  | |  | |  | |  |
| 36 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 37 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 38 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 39 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 40 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 41 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 42 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 43 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 44 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 45 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 46 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 47 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 48 | 设有如下定义: int (\*ptr)\*(); 则以下叙述中正确的是  A)ptr是指向一维组数的指针变量 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 49 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 50 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 51 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 52 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 53 | 为表示关系x≥y≥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 |  | |  | |  | |  | |  | |  | |  | |  |
| 54 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 55 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 56 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 57 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 58 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 59 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 60 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 61 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 62 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 63 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 64 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 65 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 66 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 67 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 68 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 69 | 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-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,-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 |  | |  | |  | |  | |  | |  | |  | |  |
| 70 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 71 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 72 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 73 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 74 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 75 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 76 | 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) prinft("%c\n",class[3].mane);  B) pfintf("%c\n",class[3].name[1]);  C) prinft("%c\n",class[2].name[1]);  D) printf("%^c\n",class[2].name[0]); | D |  | |  | |  | |  | |  | |  | |  | |  |
| 77 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 78 | 以下程序运行后,输出结果是 A) 1 B) 7 C)9 D)11  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);  } | C |  | |  | |  | |  | |  | |  | |  | |  |
| 79 | 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); | B |  | |  | |  | |  | |  | |  | |  | |  |
| 80 | 下列运算符中优先级最高的是\_\_\_\_\_。  A)< B)+ C)&& D)!= | B |  | |  | |  | |  | |  | |  | |  | |  |
| 81 | 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) | C |  | |  | |  | |  | |  | |  | |  | |  |
| 82 | 以下程序的 输出结果是 A) Zhang B)Zhao C) Wang D) 18  #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);  } | B |  | |  | |  | |  | |  | |  | |  | |  |
| 83 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 84 | 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] | A |  | |  | |  | |  | |  | |  | |  | |  |
| 85 | 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'}; | B |  | |  | |  | |  | |  | |  | |  | |  |
| 86 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 87 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 88 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 89 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 90 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 91 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 92 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 93 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 94 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 95 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 96 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 97 | 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 1a0 goto temp  do \_A int INT | C |  | |  | |  | |  | |  | |  | |  | |  |
| 98 | 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-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-种字符 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 99 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 100 | 下面四个选项中,均是不合法的浮点数的选项是 A) 160. B) 123 C)-.18 D)-e3  0.12 2e4.2 123e4 .234  e3 .e5 0.0 1e3 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 101 | 下面四个选项中,均是合法转义字符的选项是  A) '\'' B)'\' C)'\018' D)'\\0'  '\\' '\017' '\f' '101'  '\n' '\"' 'xab' 'xlf' | A |  | |  | |  | |  | |  | |  | |  | |  |
| 102 | 下面四个选项中,均是不正确的八进制数或十六进制数的选项是。  A) 016 B)0abc C)010 D)0a12  0x8f 017 -0x11 7ff  018 0x8 0x16 -123 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 103 | 下面不正确的字符串常量是。  A)'abc' B)"12'12" C)"0" D)"" | A |  | |  | |  | |  | |  | |  | |  | |  |
| 104 | 若有代数式3ae/bc,则不正确的C语言表达式是:  A) a/b/c\*e\*3 B)3\*a\*e/b/c  C)3\*a\*e/b\*c D)a\*e/c/b\*3 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 105 | 已知各变量的类型说明如下:  int k,a,b;  unsigned long w= 5;  double x=1.42; 则以下不符合C语言语法的表达式是: A)x%(-3) B)w+=-2 C)k=(a=2,b=3,a+b) D)a+= a-=(b=4)\*(a=3) | A |  | |  | |  | |  | |  | |  | |  | |  |
| 106 | 以下不正确的叙述是 A)在C程序中,逗号运算符的优先级最低 B)在C程序中,APH和aph是两个不同的变量 C)若a和b类型相同,在执行了赋值表达式a=b后b中的值将放人a中,而b中的值不变。 D)当从键盘输入数据时,对于整型变量只能输入整型数值,对于实型变量只能输入实型数值。 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 107 | 以下正确的叙述是: A)在C程序中,每行中只能写-条语句 B)若a是实型变量,C程序中允许赋值a=10,因此实型变量中允许存放整型数 C)在C程序中,无论是整数还是实数,都能被准确无误地表示 D)在C程序中,%是只能用于整数运算的运算符 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 108 | 以下符合C语言语法的赋值表达式是 A)d=9+e+f=d+9 B)d=9+e,f=d+9 C)d=9+e,e++,d+9 D)d=9+e++=d+7 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 109 | 已知字母A的ASCII码为10进制数65,且c2为字符型,则执行语句c2='A'+'6'-'3';后,c2中的值为 A)D B)68 C)不确定的值 D)C | A |  | |  | |  | |  | |  | |  | |  | |  |
| 110 | 为了避免在嵌套的条件语句if\_else中产生二义性,C语言规定else子句总是与( )配对。 A) 缩排位置相同的if B)其之前最近的if C) 之后最近的if D)同-行上的if | B |  | |  | |  | |  | |  | |  | |  | |  |
| 111 | 若以下变量均是整型,且num=sum=7;则执行表达式sum=num++,sum++,++num后sum的值为  A) 7 B) 8 C) 9 D) 10 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 112 | 在C语言中,int、char和short种类型数据在内存中所占用的字节数 A)由用户自己定义 B)均为2个字节 C)是任意的 D)由所用机器的机器字长决定 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 113 | 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 C)包含3个字符 D)说明不合法,c的值不确定 | A |  | |  | |  | |  | |  | |  | |  | |  |
| 114 | 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) 2.7500OO C) 3.500000 D) 0.000000 | A |  | |  | |  | |  | |  | |  | |  | |  |
| 115 | sizeof(float)是: A)1个双精度型表达式 B)1个整型表达式 C)1种函数调用D)1个不合法的表达式 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 116 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 117 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 118 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 119 | 设有说明: char w; int x; f1oat 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 120 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 121 | 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)从左起输出该字串,右补空格 B)按原字符长从左向右全部输出 C)右对齐输出该字串,左补空格 D)输出错误信息 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 122 | 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)从左起输出该字串,右补空格 B)按原字符长从左向右全部输出 C)右对齐输出该字串,左补空格 D)输出错误信息 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 123 | 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-种,它可以适用于任何-种类型的数据 C)%x 是格式符的-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 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 124 | 已有定义 int x; f1oat 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 125 | 已有定义int x; f1oat 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 126 | 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); | D |  | |  | |  | |  | |  | |  | |  | |  |
| 127 | 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-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); | D |  | |  | |  | |  | |  | |  | |  | |  |
| 128 | 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) | D |  | |  | |  | |  | |  | |  | |  | |  |
| 129 | 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; | A |  | |  | |  | |  | |  | |  | |  | |  |
| 130 | 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; | A |  | |  | |  | |  | |  | |  | |  | |  |
| 131 | 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); | B |  | |  | |  | |  | |  | |  | |  | |  |
| 132 | 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 | A |  | |  | |  | |  | |  | |  | |  | |  |
| 133 | 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 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 134 | 下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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 135 | 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) | C |  | |  | |  | |  | |  | |  | |  | |  |
| 136 | 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) | D |  | |  | |  | |  | |  | |  | |  | |  |
| 137 | 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" | C |  | |  | |  | |  | |  | |  | |  | |  |
| 138 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 139 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 140 | 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); | B |  | |  | |  | |  | |  | |  | |  | |  |
| 141 | 以下程序的运行结果是 : main()  { int m=5;  if(m++>5) printf(" %d\n",m);  e1se printf("%d\n",m--); } A)4 B)5 C)6 D)7 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 142 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 143 | 以下程序的输出结果是:  main()  {int a=100,x=10,y=20,okl=5,ok2=0;  if(x<y)  if(y!=10)  if(!okl) 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 144 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 145 | 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++; } | B |  | |  | |  | |  | |  | |  | |  | |  |
| 146 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 147 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 148 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 149 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 150 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 151 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 152 | 程序的功能是从键盘输入的字符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:  #indude "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); }  【1】 A)c=gets() B) getchar() C)c=getchar() D) scanf("%c",c) | C |  | |  | |  | |  | |  | |  | |  | |  |
| 153 | 下面程序的功能是在输入的-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==o B)a C) !a==1 D)!a | B |  | |  | |  | |  | |  | |  | |  | |  |
| 154 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 155 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 156 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 157 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 158 | 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); }  A)a=3,b=11 B )a=2,b=8 C)a=1, b=-1 D)a=4,b=9 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 159 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 160 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 161 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 162 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 163 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 164 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 165 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 166 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 167 | 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);  } } A)39 81 B)42 84 C)26 68 D) 28 70 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 168 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 169 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 170 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 171 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 172 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 173 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 174 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 175 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 176 | 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) int a[]={0}; D) int a[10]={10\*1}; | C |  | |  | |  | |  | |  | |  | |  | |  |
| 177 | 以下对二维数组a的正确说明是\_\_\_\_\_。 A) int a[3][]; B) float a(3,4); C) double a[1][4]; D) float a(3)(4); | C |  | |  | |  | |  | |  | |  | |  | |  |
| 178 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 179 | 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 C)第5行有错误 D)第7行没有错误 | A |  | |  | |  | |  | |  | |  | |  | |  |
| 180 | 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[1]); 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 181 | 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 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 182 | 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-个位置上) A)i\*m+j B)j\*p+I C)i\*m+j-1 D)i\*m+j+1 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 183 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 184 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 185 | 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 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 186 | 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 | B |  | |  | |  | |  | |  | |  | |  | |  |
| 187 | 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 | D |  | |  | |  | |  | |  | |  | |  | |  |
| 188 | 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"); | B |  | |  | |  | |  | |  | |  | |  | |  |
| 189 | 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)\O C) e D)f | D |  | |  | |  | |  | |  | |  | |  | |  |
| 190 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 191 | 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、使文件位置指针重新回到文件的开始 D、返回文件长度值 | C |  | |  | |  | |  | |  | |  | |  | |  |
| 192 | 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 |  | |  | |  | |  | |  | |  | |  | |  |
| 193 | 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));  则应填入 A)int B)int \* C)(\*int) D)(int \*) | D |  | |  | |  | |  | |  | |  | |  | |  |
| 194 | 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 |  | |  | |  | |  | |  | |  | |  | |  |