

-----一、单选题

1 结构化程序设计所规定的三种基本控制结构是 ()。

- A. 输入、处理、输出 B. 树形、网形、环形
C. 主程序、子程序、函数 D. 顺序、选择、循环

2 下列程序段不是死循环的是 ()。

- A. `int i=10;`
 `while(1)`
 `{ i=i%10+1;`
 `if(i>10) break;`
 `}`
B. `int i=1;`
 `while(i); i--;`
C. `int i;`
 `do { i=0;i++; } while(i<=0);`
D. `int i;`
 `for(i=1; ;i++);`

3 下列程序运行后的输出结果是 ()。

```
main()
{ int a=2,b=-1,c=2;
  if(a<b)
    if(b<0) c=0;
    else c++;
  printf("%d\n",c);
}
```

- A. 0 B. 1 C. 2 D. 3

4 有以下程序:

```
#define f(x) (x*x)
main()
{ int i1, i2;
  i1=f(8)/f(4); i2=f(4+4)/f(2+2);
  printf("%d, %d\n",i1,i2);
}
```

程序运行后的输出结果是 ()。

- A. 4, 3 B. 4, 4 C. 64, 28 D. 64, 64

5 若以下选项中的变量已正确定义, 则正确的赋值语句是 ()。

- A. `x3=0x12;` B. `1+2=x2;` C. `x1=26.8%3;` D. `x4=1+2=3;`

6 下列程序中, while 循环的循环次数是 ()。

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

A. 1 B. 10 C. 6 **D. 死循环，不能确定次数**

7 下列程序运行时，若输入 1 2 3 4 5 0<回车>，则输出结果是（ ）。

```
#include <stdio.h>
main()
{ int s;
  scanf("%d",&s);
  while(s>0)
  { switch(s)
    { case 1:printf("%d",s+5);
      case 2:printf("%d",s+4); break;
      case 3:printf("%d",s+3);
      default:printf("%d",s+1);break;
    }
    scanf("%d",&s);
  }
}
```

A. 66666 B. 66656 **C. 6566456** D. 6666656

8 若有定义：int aa[8];，则不能代表数组元素 aa[1]地址的是（ ）。

A. &aa[0]++ B. &aa[1] C. &aa[0]+1 D. aa+1

9 下列程序运行后的输出结果是（ ）。

```
main()
{ char arr[2][4];
  strcpy(arr,"you");
  strcpy(arr[1],"me");
  arr[0][3]='&';
  printf("%s\n",arr);
}
```

A. you **B. you&** C. you&me D. youme

10 下面关于全局变量的描述中，错误的是（ ）。

- A. 所有在函数体外定义的变量都是全局变量
- B. 全局变量第一次被引用时，系统为其分配内存**
- C. 全局变量可以和局部变量的名称相同
- D. 全局变量直到程序结束时才被释放

11 有以下程序：

```
#include <stdio.h>
int fun()
{ static int x=1;
  x*=2;
  return x;
}
main()
{ int i,s=1;
  for(i=1;i<=3;i++)
    s*=fun();
  printf("%d\n",s);
}
```

```
}

```

程序运行后的输出结果是 ()。

- A. 0 B. 10 **C. 64** D. 30

12 下面函数的功能是 ()。

```
fun(char *p2, char *p1)
{ while((*p2=*p1)!='\0')
  { p1++; p2++; }
}
```

- A. 将 p1 所指字符串复制到 p2 所指内存空间
B. 将 p1 所指字符串的地址赋给指针 p2
C. 对 p1 和 p2 两个指针所指字符串进行比较
D. 检查 p1 和 p2 两个指针所指字符串中是否有 '\0'

13 有以下函数：

```
char *fun(char *p)
{ return p;}
```

该函数的返回值是 ()。

- A. 无确定的值 B. 形参 p 自身的地址值
C. 一个临时存储单元的地址 **D. 形参 p 中存放的地址值**

14 设有如下说明：

```
typedef struct { int n; char c; double x;} STD;
```

则以下选项中，能正确定义结构体数组并赋初值的语句是 ()。

- A. struct tt[2]={1,'A',2,'B'};
B. STD tt[2]={1,"A",62},{2,"B",75};
C. STD tt[2]={1,'A',62},{2,'B',75};
D. struct tt[2]={1,"A",62.5},{2,"B",75.0};

15 函数调用语句：fseek(fp,-10L,2);的含义是将文件位置指针 ()。

- A. 移动到距离文件尾 10 个字节处 **B. 从当前位置向文件尾方向移动 10 个字节**
C. 移动到距离文件头 10 个字节处 D. 从当前位置向文件头方向移动 10 个字节

16 构成 C 程序的基本单位是 ()。

- A. 函数** B. 过程 C. 子程序 D. 子例程

17 设有定义：int a=2,b=3,c=4;，则以下选项中值为 0 的表达式是 ()。

- A. a && b B. (a<b)&& !c||1
C. (!a==1)&&(!b==0) D. a||(b+b)&&(c-a)

18 执行以下程序后，a,b 的值分别为 ()。

```
main()
{ int a,b,k=4,m=6,*p1=&k,*p2=&m;
  a=p1==&m;
  b=(*p1)/(*p2)+7;
  printf("a=%d\n",a);
  printf("b=%d\n",b);
}
```

- A. -1,5 B. 1,6 **C. 0,7** D. 4,10

19 若有定义: `int a=1,b=10;`, 执行下列程序段后, `b` 的值为 ()。

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

A. 9 **B. 8** C. -1 D. -2

20 下列程序段的输出结果是 ()。

`int n='c';`

`switch(n++)`

`{ default: printf("error"); break;`

`case 'a': case 'A': case 'b':`

`case 'B': printf("good"); break;`

`case 'c': case 'C': printf("pass");`

`case 'd': case 'D': printf("warn");`

`}`

A. passwarn **B. error** C. warn D. pass

21 若有定义: `int x,y;`, 则循环语句 `for(x=0,y=0;(y!=123)||(x<4);x++);` 的循环次数为 ()。

A. 不确定次 B. 无限次 C. 4 次 **D. 3 次**

23 以下语句或语句组中, 能正确进行字符串赋值的是 ()。

A. `char *sp="right!";` B. `char s[10]; s="right!";`

C. `char s[10]; *s="right!";` D. `char *sp; *sp="right!";`

22 有下面的程序片段, 则对数组元素的错误引用是 ()。

`int a[12]={0},*p[3],**pp,i;`

`for(i=0;i<3;i++)`

`p[i]=&a[i*4];`

`pp=p;`

A. `pp[0][1]` B. `a[10]` **C. `p[3][1]`** D. `*(*(p+2)+2)`

23 以下叙述中正确的是 ()。

A. 形参可以使用的存储类说明符与局部变量完全相同

B. 全局变量说明为 `static` 存储类, 其作用域将被扩大

C. 任何存储类的变量在未赋初值时, 其值都是不确定的

D. 局部变量说明为 `static` 存储类, 其生存期将得到延长

24 若有如下定义和说明:

`int (*pf)();`

`int fun();`

则正确的赋值语句是 ()。

A. `pf=fun;` B. `*pf=fun;` C. `pf=&fun;` D. `pf=fun();`

25 下列程序的运行结果是 ()。

`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);`


```
}

```

A. 6 3 **B. 3 6** C. 编译出错 D. 0 0

26 对于下面的定义，不正确的叙述是（ ）。

```
union data

```

```
{ char c;int i;float f;}a;

```

A. 变量 a 所占内存的长度等于成员 f 的长度

B. 变量 a 的地址与它的各成员的地址相同

C. 变量 a 可以作函数的参数

D. 不可以在定义时对 a 初始化

27 设有以下结构体类型数组的定义，且数组 mini 的 10 个元素都已赋值，若要将这些元素写到 fp 所指向文件中，以下不正确的形式是（ ）。

```
struct abc { int a;char b;float c[4];} mini[10];

```

A. fwrite(mini,10*sizeof(struct abc),1,fp);

B. for(i=0;i<10;i++)

fwrite(mini,sizeof(struct abc),1,fp);

C. fwrite(mini,5*sizeof(struct abc),2,fp);

D. fwrite(mini,sizeof(struct abc),10,fp);

28 设有定义：int n=0,*p=&n,**q=&p，则下列选项中正确的赋值语句是（ ）。

A. p=1; B. *q=2; C. q=p; D. *p=5;

29 C 语言中，while 和 do-while 循环的主要区别是（ ）。

A. while 的循环控制条件比 do-while 的循环控制条件严格

B. do-while 的循环体至少无条件执行一次

C. do-while 允许从外部转到循环体内

D. do-while 的循环体可以是复合语句

30 下列程序运行后的输出结果是（ ）。

```
main()

```

```
{ int i,n=0;

```

```
  for(i=2;i<5;i++)

```

```
  { do

```

```
    { if(i%3) continue;

```

```
      n++;

```

```
    } while(!i);

```

```
    n++;

```

```
  }

```

```
  printf("n=%d\n",n);

```

```
}

```

A. n=5 B. n=2 C. n=3 **D. n=4**

31 以下选项中与 if(a==1)a=b; else a++;语句功能不同的 switch 语句是（ ）。

A. switch(a)

B. switch(a==1)

```
  { case 1:a=b;break;

```

{ case 0:a=b;break;

```
    default:a++;

```

case 1:a++;

```
  }

```

}

C. switch(a)

D. switch(a==1)

```

{ default:a++;break;
  case 1:a=b;
}

```

```

{ case 1:a=b;break;
  case 0:a++;
}

```

32 若二维数组 y 有 m 列，则排在 $y[i][j]$ 前的元素个数为 ()。

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

33 有定义语句: `int (*prt)[3]`，则标识符 `prt` ()。

- A. 是一个指向整型变量的指针
 B. 是一个指针数组名，每个元素都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. 定义不合法

34 以下程序执行后的输出结果是 ()。

```

#include "string.h"
main()
{ char *p[10]={ "abc","aabdfg","dcdbe","abbd","cd"};
  printf("%d\n",strlen(p[4]));
}

```

- A. 5 B. 4 C. 3 **D. 2**

35 下面关于局部变量的描述中，错误的是 ()。

- A. 函数的形式参数是局部变量
B. static 型局部变量的作用域是当前文件
 C. 在复合语句中定义的变量在本复合语句中有效
 D. 不同函数中可以定义相同名称的局部变量

36 有以下程序:

```

#include <stdio.h>
void fun(char *c,int d)
{ *c=*c+1;
  d=d+1;
  printf("%c,%c,",*c,d);
}
main()
{ char b='a',a='A';
  fun(&b,a);
  printf("%c,%c\n",b,a);
}

```

程序运行后的输出结果是 (**bBAb**)。

- A. a, B, B, a B. b, B, B, A C. b, B, b, A D. a, B, a, B

37 下面函数的类型是 ()。

```

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

```

- A. 与参数 x 的类型相同 B. void **C. int** D. 无法确定

38 有以下定义和语句:

```

struct workers
{ int num;char name[20];char c;
  struct { int day; int month; int year; } s;
}

```

```
};
```

```
struct workers w,*pw=&w;
```

能给 w 中 year 成员赋 1980 的语句是 ()。

A. *pw.year=1980; **B. w.s.year=1980;** C. pw->year=1980; D. w.year=1980;

39 标准库函数 fputs(p1,p2)的功能是 ()。

A. 从 p1 指向的文件中读取一个字符串存入 p2 开始的内存

B. 从 p2 指向的文件中读取一个字符串存入 p1 开始的内存

C. 从 p2 开始的内存中读取一个字符串存入 p1 指向的文件

D. 从 p1 开始的内存中读取一个字符串存入 p2 指向的文件

40 下列程序的输出结果是 ()。

```
main()
```

```
{ int a[5]={2,4,6,8,10},*p,**k;
```

```
  p=a;
```

```
  k=&p;
```

```
  printf("%d",*(p++));
```

```
  printf("%d\n",**k);
```

```
}
```

A. 24 B. 22 C. 4 D. 46

41 若已定义的函数有返回值，则以下关于该函数调用的叙述中错误的是 ()。

A. 函数调用可以作为独立的语句存在

B. 函数调用可以作为一个函数的实参

C. 函数调用可以出现在表达式中

D. 函数调用可以作为一个函数的形参

42 下列程序运行后的输出结果是 ()。

```
void swap1(int c0[],int c1[])
```

```
{ int t;t=c0[0];c0[0]=c1[0];c1[0]=t;}
```

```
void swap2(int *c0,int *c1)
```

```
{ int t;t=*c0; *c0=*c1; *c1=t;}
```

```
main()
```

```
{ int a[2]={3,5},b[2]={3,5};
```

```
  swap1(a,a+1);
```

```
  swap2(&b[0],&b[1]);
```

```
  printf("%d %d %d %d\n",a[0],a[1],b[0],b[1]);
```

```
}
```

A. 3 5 5 3 B. 3 5 3 5 C. 5 3 5 3 D. 5 3 3 5

43 设 m,n,a,b,c,d 的值均为 0，执行(m=a==b)||(n=c==d)后，m,n 的值是 ()。

A. 0,1 **B. 1,0** C. 0,0 D. 1,1

44 已知 i、j、k 为 int 型变量，若从键盘输入：1，2，3<回车>，使 i 的值为 1、j 的值为 2、k 的值为 3，以下选项中正确的输入语句是 ()。

A. scanf("%2d%2d%2d",&i,&j,&k);

B. scanf("%d,%d,%d",&i,&j,&k);

C. `scanf("%d %d %d",&i,&j,&k);`

D. `scanf("i=%d,j=%d,k=%d",&i,&j,&k);`

41 与 `for(;;)` 等价的是 ()。

A. `while(0)` B. `continue` C. `break` D. `while(1)`

45 有以下程序:

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

```
main( )
```

```
{ char i;
```

```
  for(;;(i=getchar())!='\n');
```

```
  { switch(i-'a')
```

```
    { case 0: putchar(i);
```

```
      case 1: putchar(i+1); break;
```

```
      case 2: putchar(i+2);
```

```
      case 3: break;
```

```
      default: putchar(i); break;
```

```
    }
```

```
  }
```

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

```
}
```

输入下列数据后, 程序的输出结果是 ()。

abcde<回车>

A. abcde B. abbde C. **abcee** D. abcedd

46 有以下程序段:

```
int n=0,p;
```

```
do{ scanf("%d",&p); n++; }while(p!=12345&& n<3);
```

此处 do-while 循环的结束条件是 ()。

A. p 的值不等于 12345 并且 n 的值小于 3

B. p 的值等于 12345 并且 n 的值大于等于 3

C. p 的值不等于 12345 或者 n 的值小于 3

D. **p 的值等于 12345 或者 n 的值大于等于 3**

47 设有程序段:

```
#define N 3
```

```
#define M N+2
```

```
float a[M][N];
```

则数组 a 的元素个数和最后一个元素分别为 ()。

A. **15,a[4][2]** B. 15,a[3][5] C. 8,a[4][2] D. 15,a[5][3]

48 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. **一个返回值为指针型的函数名**

D. 一个用于指向函数的指针变量

49 有以下程序:

```
int a=2;
```

```
int f(int n)
```



```

{ static int a=3;
  int t=0;
  if(n%2){ static int a=4; t+=a++;}
  else { static int a=5; t+=a++;}
  return t+a++;
}
main()
{ int s=a,i;
  for(i=0;i<3;i++) s+=f(i);
  printf("%d\n",s);
}

```

程序运行后的输出结果是 ()。

- A. 26 B. 28 **C. 29** D. 24

50 下面函数的功能是 ()。

```

fun(char *p2, char *p1)
{ while((*p2=*p1)!='\0')
  { p1++; p2++; }
}

```

- A. 对 p1 和 p2 两个指针所指字符串进行比较
 B. 将 p1 所指字符串的地址赋给指针 p2
C. 将 p1 所指字符串复制到 p2 所指内存空间
 D. 检查 p1 和 p2 两个指针所指字符串中是否有 '\0'

51 设有说明语句: struct stu{ int a; float b; }stutype;, 则叙述不正确的是 ()。

- A. stutype 是用户定义的结构体类型名**
 B. struct stu 是用户定义的结构体类型
 C. stuct 是结构体类型的关键字
 D. a 和 b 都是结构体成员名

52 下面对枚举类型的叙述, 不正确的是 ()。

- A. 一个整数可以直接赋给一个枚举变量** B. 定义枚举类型用 enum 开头
 C. 枚举常量的值是一个常数 D. 枚举值可以用来作判断比较

53 以“w+”方式打开一个文件, 下列描述中正确的是 ()。

- A. 文件打开时, 原有文件内容不被删除, 位置指针移动至文件末尾, 可进行写和读操作
B. 文件打开时, 原有文件内容被删除, 可进行写和读操作
 C. 文件打开时, 原有文件内容不被删除, 位置指针移动至文件开头, 可进行重写和读操作
 D. 以上说法都不正确

54 计算机高级语言程序的运行方法有编译执行和解释执行两种, 以下叙述中正确的是 ()。

- A. C 语言程序既可以编译执行又可以解释执行 B. C 语言程序仅可以解释执行
C. C 语言程序仅可以编译执行 D. 以上说法都不对

55 以下定义语句中正确的是 ()。

- A. int a,*p=&a;** B. int a,*p=a; C. int *p=10; D. int *p=&a,a;

56 若 x 和 y 代表整型数，以下表达式中不能正确表示数学关系 $|x-y|<10$ 的是 ()。

- A. $\text{abs}(x-y)<10$ B. $x-y>-10\&\&x-y<10$
C. $!(x-y)<-10\|!(y-x)>10$ D. $(x-y)*(x-y)<100$

57 以下对二维数组的定义中正确的是 ()。

- A. $\text{int a}[[3]]=[1, 2, 3, 4, 5, 6];$ B. $\text{int a}[[3];$
C. $\text{int a}[4][]=\{1, 2, 3, 4, 5, 6\};$ D. $\text{int a}[[]=\{\{1,2,3\},\{4,5,6\}\};$

58 若有如下定义，则 b 的值是 ()。

- $\text{int a}[10]=\{1,2,3,4,5,6,7,8,9,10\}, *p=\&a[3], b=p[5];$
A. 9 B. 6 C. 8 D. 5

59 下面程序段的运行结果是 ()。

```
char a[]="lanuage",*p;  
p=a;  
while(*p!='u') {printf("%c",*p-32); p++; }  
A. LANGUAGE                      B. language                      C. LAN                      D. langUAGE
```

60 若有语句 $\text{char s1}[10], \text{s2}[10]="books";$ ，则能将字符串 books 赋给数组 s1 的语句是 ()。

- A. $\text{s1}="books";$ B. $\text{strcpy}(\text{s2}, \text{s1});$ C. $\text{s1}=\text{s2};$ D. $\text{strcpy}(\text{s1}, \text{s2});$

61 下列对静态局部变量的叙述哪一个是不正确的 ()。

- A. 静态局部变量在整个程序运行期间都不释放
B. 静态局部变量是在编译时赋初值的，故它只能被初始化一次
C. 在一个函数中定义的静态局部变量可以被另一个函数使用
D. 数值型静态局部变量的初值默认为 0

62 有以下程序：

```
#include <stdio.h>  
int f(int n);  
main()  
{ int s;  
  s=f(4); printf("%d\n",s);  
}  
int f(int n)  
{ int s;  
  if(n>0) s=n+f(n-1);  
  else s=0;  
  return s;  
}
```

63 程序运行后的输出结果是 ()。

- A. 4 B. 14 C. 10 D. 6

64 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

65 若有语句 `enum color {red=-1,yellow,blue,white};`, 则 `blue` 的机内值是 ()。

- A. 0 B. 1 C. 2 D. 3

66 C 语言中, 库函数 `fgets(str,n,fp)` 的功能是 ()。

- A. 从 `fp` 所指向的文件中读取长度不超过 `n-1` 的字符串存入 `str` 开始的内存
B. 从 `fp` 所指向的文件中读取长度为 `n` 的字符串存入 `str` 开始的内存
C. 从 `fp` 所指向的文件中读取 `n` 个字符串存入 `str` 开始的内存
D. 从 `str` 开始的内存读取至多 `n` 个字符存入 `fp` 所指向的文件

67 要把高级语言编写的源程序转换为目标程序, 需要使用 ()。

- A. 编辑程序 B. 编译程序 C. 诊断程序 D. 驱动程序

68 下列程序运行后的输出结果是 ()。

```
main()
{ int a=16,b=22,m=0;
  switch(a%3)
  { case 0:m+=1;break;
    case 1:m+=2;
    switch(b%3)
    { case 1:m+=4;break;
      default:m+=3;
    }
  }
  printf("%d\n",m);
}
```

A. 6 B. 3 C. 5 D. 2

69 下列程序中的 `for` 循环执行的次数是 ()。

```
#define N 2
#define M N+1
#define NUM 2*M+1
main()
{ int i;
  for(i=1;i<=NUM;i++) printf("%d\n",i);
}
```

A. 5 B. 6 C. 7 D. 8

70 以下程序中，while 循环的执行次数是（ ）。

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

A. 1 B. 6 C. 死循环，不能确定次数 D. 10

71 下列程序的输出结果是（ ）。

```
main()
{ char a[10]={9,8,7,6,5,4,3,2,1,0},*p=a+5;
  printf("%d",*--p);
}
```

A. 运行出错 B. a[4]的地址 C. 3 D. 5

72 将 p 定义为指向含 4 个元素的一维数组的指针变量，正确语句为（ ）。

A. int p[4]; B. int *p[4]; C. int (*p)[4]; D. int **p[4];

73 以下程序的输出结果是（ ）。

```
main()
{ int m[][3]={1,4,7,2,5,8,3,6,9};
  int i,j,k=2;
  for(i=0;i<3;i++)
  { printf("%d ",m[k][i]);}
}
```

A. 4 5 6 B. 2 5 8 C. 7 8 9 D. 3 6 9

74 有以下语句，则对 a 数组元素的引用不正确的是($0 \leq i \leq 9$)（ ）。

```
int a[10]={0,1,2,3,4,5,6,7,8,9},*p=a;
```

A. a[p-a] B. *(&a[i]) C. p[i] D. *(*a+i)

75 设有如下函数定义：

```
int fun(int k)
{ if(k<1) return 0;
  else if(k==1) return 1;
  else return fun(k-1)+1;
}
```

若执行调用语句：n=fun(3);，则函数 fun 总共被调用的次数是（ ）。

81 以下能正确定义且赋初值的语句是 ()。

A. char n1=n2='a'; B. char c=32; C. char a,b=a; D. char x="A";

82 若希望当 A 的值为奇数时，表达式的值为真，A 的值为偶数时，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)

83 下列程序运行后的输出结果是 ()。

```
main()
{ int m=14,n=63;
  while(m!=n)
  {   while(m>n) m=m-n;
      while(m<n) n=n-m;
  }
  printf("%d\n",m);
}
```

A. 7 B. 63 C. 14 D. 0

84 下列程序运行后的输出结果是 ()。

```
#define F(X,Y) (X)*(Y)
main()
{ int a=3,b=4;
  printf("%d\n",F(a++,b++));
}
```

A. 20 B. 16 C. 15 D. 12

85 下列条件语句中，功能与其他语句不同的是 ()。

A. if(a) printf("%d\n",x); else printf("%d\n",y);
B. if(a==0) printf("%d\n",x); else printf("%d\n",y);
C. if(a==0) printf("%d\n",y); else printf("%d\n",x);
D. if(a!=0) printf("%d\n",x); else printf("%d\n",y);

86 下列程序执行后的输出结果是 ()。

```
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]);
}
```

A. 6 B. 9 C. 3 D. 随机数