1、请解释“func”为何种类型，这种类型的作用什么，变量ttt 的值是多少？

typedef int (\*func)(int, int\*);

int xxx(int a, int \*p)

{

return a + \*p;

}

int dowork(func aaa, int bbb, int \*ccc)

{

return aaa(bbb, ccc);

}

int sss = 4;

int ttt = dowork(&xxx, 3, &sss);

1. 若有定义inta[3][3],有哪几种方式可以表示a[1][1]。

3、用变量a给出下面的定义

a) 一个整型数（An integer）

b) 一个指向整型数的指针（A pointer to an integer）

c) 一个指向指针的的指针，它指向的指针是指向一个整型数（A pointer to a pointer to an integer）

d) 一个有10个整型数的数组（An array of 10 integers）

e) 一个有10个指针的数组，该指针是指向一个整型数的（An array of 10 pointers to integers）

f) 一个指向有10个整型数数组的指针（A pointer to an array of 10 integers）

g) 一个指向函数的指针，该函数有一个整型参数并返回一个整型数（A pointer to a function that takes an integer as an argument and returns an integer）

h) 一个有10个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数（ An array of ten pointers to functions that take an integer argument and return an integer ）

4、 int a=12; int\*p,\*q，\*r;

p=&a;

r=p+1;

q=p++;

printf(“%2d%2d\n”,q-p,r-p);

printf(“%5d\n”,(char\*)p-(char\*)q);

输出结果是多少？

5、下面程序执行的结果？为什么会是这个结果？

int main()

{

char a[1000];

int i;

for(i = 0; i < 1000; i++)

{

a[i] = -1-i;

}

printf(“%d\n”, strlen(a));

return 0;

}

6、Int main{

Char a[5] = {‘A’,’B’,’C’,’D’}；

Char (\*p1)[5] = &a;

Char (\*p2)[5] = a;

Return 0;

}

问题： 1.p1和p2 的使用，哪个正确？

1. P1+1的值会是什么？ P2+1的值是什么？

7、请问下面程序有什么错误?

int a[60][250][1000],i,j,k;

for(k=0;k<=1000;k++)

for(j=0;j<250;j++)

for(i=0;i<60;i++)

a[i][j][k]=0;

8、数组定义为“ int a [ 4 ] ; ”，表达式 ( ) 是错误的。   
A.\*a   
B.a [ 0 ]   
C.a   
D.a++

9、打印一个N\*N的方阵，N为每边字符的个数（ 3〈N〈20 ），要求最外层为“X”，第二层为“Y”，从第三层起每层依次打印数字0，1，2，3，...   
例子：当N =5，打印出下面的图形：   
X X X X X  
X Y Y Y X   
X Y 0 Y X   
X Y Y Y X   
X X X X X

10、已知一个数组table，用一个宏定义，求出数据的元素个数。

#define NUM(a) while(\*a != ‘\0’) count++;

11、若定义：int a[2][3]= {0,2,4,6,8,10},以下说法正确的是：

1. \*（a + 1）为元素6的地址
2. \*(a[1] + 1)的值为2
3. a[0]与a相同
4. \*\*（a+1）+2的值为8
5. a[1][2]的值为10

12、用函数指针实现a,b中取最大值

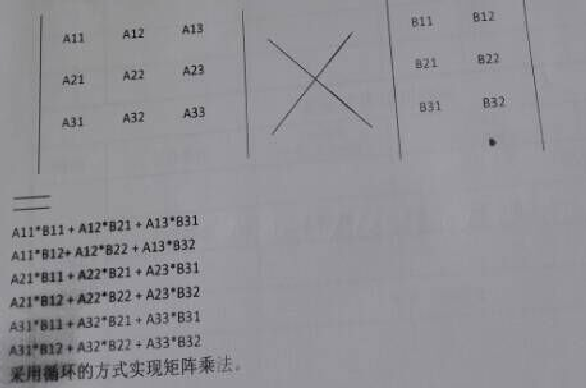
1. 函数指针有什么作用
2. 指针在任何情况下都可以进行> ,< ,<=,>=,==运算，这句话对不对
3. 设int arr[] = {6,7,8,9,10}

Int \*ptr = arr;

\*(ptr++)+ = 23;

\*ptr =? \*(++)ptr = ?

16、



17、下列程序的运行结果是( )

main( )   
{ int a[][4]={1，3，5，7，9，11，13，15，17，19，21，23｝;

　　int (\*p)[4]，i=2，j=1;

　　p=a;

　　printf(″%d\n″，\*(\*(p+i)+j));

}   
A.9  B.11  C.17  D.19

18、 Char\* pa[3] = {“ABC”,”DEF”,”GHI”};

Char\*\* p = pa；

p ++；

printf(“%s\n”,p);

打印结果是什么？

19、int a[5]; 假设a的首地址是0xff15；

求 a+1 &a+1 &(a+1) sizeof(a)/sizeof(int)

20、某无序数组arry[]中，将数组中的奇数和偶数分开，并排序输出到两个数组中。如int arry[] = {1,2,3,4,56,43,22,12,……};

21、给定一个数组a[N]，数组的各个元素值为随机的unsigned int型，另有一空数组b[n]，先将a中最大值放至b的最中间（若N为奇数，则直接放至最中间，若N为偶数，则放至中间的右边一个），然后依次将a中剩下的最大数放至b的中间的左边、右边交替进行，全部防止完成后，输出数组b。

如a[10]={1,2,3,4,5,6,7,8,9,10};则最后b[10]={1,3,5,7,9,10,8,6,4,2};

22、编程实现在屏幕上打印出某种数阵，

如N=4，则打印出：

1 12 11 10

2 13 16 9

3 14 15 8

4 5 6 7

如N=5，则打印出：

1 16 15 14 13

2 17 24 23 12

3 18 25 22 11

4 19 20 21 10

5 6 7 8 9

23、

int main()    
{    
    char \*str[]={"welcome","to","wuhan","farsight"};    
    char \*\*p=str+1;    
    str[0]=(\*p++)+2;    
    str[1]=\*(p+1);    
    str[2]=p[1]+3;    
    str[3]=p[0]+(str[2]-str[1]);  
      
    printf("%s\n", str[0]);    
    printf("%s\n", str[1]);    
    printf("%s\n", str[2]);    
    printf("%s\n", str[3]);    
      
    return 0;  
  }  
  NULL  
  farsight  
  sight  
  ht  
    
解答：  
a.声明一个指针数组，而str的值就是数组的起始地址值，str的值被默认解析成char \*\*变量的值。  
b.声明了一个2级指针，char\*\*p=str+1;由于str是char\*\*型，即str指向的类型是char \*型的，所以p的值是（int）str+sizeof（char\*），  
  显然p指向了str的第二个元素即str[1]的地址。  
c.(\*p++)+2;，由于后置运算符++优先级比\*高，所以先进行++运算，p的值变为（int）p+sizeof（char\*），  
   显然此时p指向了str[2]的地址，再进行\*运算，由于后置是++返回未变化的p的值，取到了str[1]的值，即指向字符串"to"的指针的值，  
   由于\*p类型是char\*的所以](\*p++)+2;最终str[0]指向了“to”结尾的'\0'所以输出str[0]时为空。  
d.str[1]=\*(p+1);，显然str[1]指向了字符串"farsight"的首地址。str[1]输出字符串farsight  
e.很显然str[2];输出sight  
f.p指向了str[2]的地址，把&str[2]~&str[2]+3里的数据修改成了字符创"jing"的首地址。str[2]-str[1] = 3，所以最终str[3]; 输出ht。