1. **选择题（每题2分）  
   1.给出以下定义:  
        char X[ ]= "abcdefg";  
        char Y[ ]= {'a','b','c','d','e','f','g'};  
        则正确的叙述为(B )  
        A) 数X和数组Y等价              B) 数组X和数组Y的长度相同  
        C) 数组X的长度大于数组Y的长度    D) 数组X的长度小于数组Y的长度**

**2.  
  void example(char acHello[])  
  {  
      printf("%d", sizeof(acHello));  
      return;  
  }  
  void main()  
  {  
      char acHello[] = "hello";  
      example(acHello);//数组名称作参数，传的是地址，一个地址占四个字节  
      return;  
  }  
的输出是 C  
A 4     B 5         C 6      D不确定**

**3. 若已定义：int a[9],\*p=a,并在之后的语句并未改变p的值，不能表示a[1]的地址的表达式是\_\_\_\_C\_\_\_\_\_\_。**

**Ap+1 B a+1 C a++ D ++p**

**4.以下叙述中不正确的是( D )  
     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(复合语句指函数中的成对括号构成的代码)**

**5.设有如下定义:  
    unsigned long pulArray[] = {6, 7, 8, 9, 10};  
    unsigned long \*pulPtr;  
    则下列程序段的输出结果为(  D )  
    pulPtr = pulArray;  
    \*(pulPtr + 2) += 2;  
    printf ("%d,%d/n", \*pulPtr, \*(pulPtr + 2));  
    A)8,10         B)6,8            C)7,9             D)6,10**

**6. struct stu  
    {   
        int      num;  
        char     name[10];  
        int      age;  
    };  
    void fun(struct stu \*p)  
    {   
        printf("%s/n", (\*p).name);  
        return;  
    }  
    void main()  
    {  
      struct  stu   students[3]={ {9801,"Zhang",20},  
                                  {9802,"Wang",19},  
                                  {9803,"Zhao",18} };  
      fun(students + 2);  
      return;  
    }  
    输出结果是(  B )  
    A) Zhang        B)Zhao        C) Wang           D) 18**

**7.在函数内部定义的变量(静态变量、寄存器变量等特殊变量除外)的内存是在栈内存中，所以在定义函数内部的变量的时候，一定要保证栈不能够溢出。如果临时变量  
 占用空间较大，应该使用内存申请的方式，这样该变量指向的内存就是在堆内存中了。 A  
 A. 正确  B. 错误**

**8.设有以下宏定义:  
     #define      N         3  
     #define      Y(n)      ((N+1)\*n) /\*这种定义在编程规范中是严格禁止的\*/  
     则执行语句:z = 2 \* (N + Y(5 + 1));后,z的值为(  A )  
     A) 出错        B) 42           C) 48            D)54**

**9.int \*(\*ptr)();   
    则以下叙述中正确的是( D  )  
    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是指向函数的指针,该函数的返回值是指向int型数据的指针**

**10.声明一个指针，其所指向的内存地址不能改变，但内存中的值可以被改变，正确的是（B）**

**A.constintconst \*x = &y;**

**B.int \* const x = &y;**

**C.constint \*x = &y;**

**D.constint \* const x = &y;**

**11.（（a>b）? a : b）代表什么含义：B**

**A：代表a,b之中值较小的数**

**B：代表a,b之中值较大的数**

**C：代表a是否比b大**

**D：语法错误**

**12.若有函数max(a,b),并且函数指针变量p已经指向函数,当调用该函数时正确的调用方法是（多选）\_A B\_\_\_\_\_.**

**A：(\*p)(a,b)**

**B：p(a,b)**

**C: (p)max(a,b)**

**D: (\*p)max(a,b)**

**13.若有以下定义:**

**char a; intb;float c; double d; 则表达式a\*b+d-c值的类型为(  D  )。**

**A． float B. intC. charD. double**

**14.递归调用非常危险，可能导致很多问题，即使程序编写没有逻辑错误，也可能导致下面哪种现象的发生（ B ）。**

**(A) 死循环**

**(B) 栈溢出**

**(C) 内存泄漏**

**15.．char 占1个字节，int占4个字节，double 占8个字节。**

**以下定义**

**Struct STUDENT**

**{**

**char name[10];**

**int age;**

**char sex;**

**float score;**

**}data;**

**sizeof(data) = ?（  D）**

**(A)  8 (B)  13 (C)  16 (D) 24**

**二.填空题（每题2分）  
1.写一个“标准”宏，这个宏输入两个参数并返回较小的一个 #Define MAX((a),(b)) (a)< (b) ? (a) : (b) .**

**2. 堆栈溢出通常是由\_\_\_\_程序空间不足、数组下标越界、指针非法访问或递归过深\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_导致的.**

**//3. .Internet采用哪种网络协议\_\_\_\_\_ \_\_\_\_\_\_\_\_\_,该协议的主要层次结构\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_.**

**4.gcc的编译步骤按先后顺序分别为\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_.**

**5.对于一个频繁使用的短小函数,在C语言中应用\_ 宏定义\_\_\_\_\_实现,在C++中应用\_\_\_\_\_\_\_实现**

**三.解答题（每题10分）**

**1.const关键字的作用？**

**答：使变量的值不可变**

**2.程序的内存分配分为哪几个部分，分别是什么？**

**答：栈区分配、静态区分配、堆区分配**

**3.statac全局变量、局部变量、函数与普通全局变量、局部变量、函数  
static全局变量与普通的全局变量有什么区别？static局部变量和普通局部变量有什么区别？static函数与普通函数有什么区别？**

**答：static全局变量与普通的全局变量，两者生命周期都为程序运行的整个过程，但作用域不同，前者为当前文件，后者为整个程序的多个文件。**

**static局部变量和普通局部变量，前者不可被其他文件引用，后者可以；调用结束后，前者的值保留，后者不保留；内存分配前者静态区，后者栈区。**

**static函数与普通函数，前者不能被其他文件引用，后者可以。**

**abccade**

**dgcadde**

**四.编程题（每题10分）**

**1.****编程实现：找出两个字符串中最大公共子字符串,如"abccade","dgcadde"的最大子串为"cad"**

**答：#include<stdio.h>**

**int main()**

**{**

**char s1[100]={0};**

**char s2[100]={0};**

**char \* p=s1;**

**char \* q =s2;**

**char \* t = NULL;**

**char \* t1 = NULL;**

**int cnt = 0;**

**int max = 0 ;**

**int i = 0;**

**gets(s1);**

**gets(s2);**

**while(\*p!='\0')**

**{**

**while(\*q!='\0')**

**{**

**if(\*p==\*q)**

**{**

**t=p;**

**while(\*t==\*q&&\*q!='\0')**

**{**

**t++;q++;**

**cnt++;**

**}**

**if(cnt>max)**

**{**

**t1=p;**

**max = cnt;**

**}**

**t=p;**

**}**

**cnt=0;**

**q++;**

**}**

**q=s2;**

**p++;**

**}**

**printf("%d\n",max);**

**while (i<max)**

**{**

**putchar(\*t1++);**

**i++;**

**}**

**return 0;**

**}**

**2.实现单链表排序（冒泡排序|快速排序）**

**3.实现两个双向不循环链表（数据域保存整型值），按整型数的大小拼接，从小到大排序（冒泡排序|快速排序）。**