一、选择题(每题 1 分，总计 8 分)

1．在 Win2K 下，32 位 C 语言程序中有如下类型的数据，其在存储器中占的字节数为（D ）

```
struct lpa
{
  char a:1;
  char b:5;
  int c;
  char m;
};
```

A．6　B．7　C．9　D．12

2．下列代码中对指针的操作正确的有（ B ）

```
#define    MAX      10
A.
int fun( void)
{
  char *a=NULL;
char *b=NULL;
  a=malloc( sizeof(char)* MAX);
  b=a;
  free(a);
  if(b!=NULL)
*b=  0  ;
……
  return 0;
}
B.
int fun(void)
{
  char *a=NULL;
  char *b=NULL;
  a=malloc( sizeof(char)* MAX);
  b=a;
  ……
  free(a);
  a=NULL;
  b=NULL;
  ……
  return 0;
}
C.
int fun(void)
{
  char *a=NULL;
  char *b=NULL;
  unsigned int i;
```

```
  a=malloc( sizeof(char)* MAX);
  b=a;
  for(i=0;i<MAX;i++)
*(b++)=i;
  free(b);
  free(a);
  return 0;
}
```

D.

```
int fun(void)
{
  char *a=NULL;
  int *b=NULL;
  a=malloc( sizeof(char)* MAX);
  b=a+9;
  *b=15;
  ……
  return b;
}
```

3. 有如下代码

```
typedef struct ptrblock {
char *ptr;
char  name[10];
} Node_t;

main()
{
  Node_t *p;
  p = (PB *)malloc(sizeof(Node_t));
  p->ptr = malloc(10);
  ……
  [内存释放语句]
  }
```

请问下面哪段代码正确完成了内存释放（C ）

A.

```
free(p);
free(p->ptr);
```

B.

```
free(p->ptr);
free(p->name);
free(p);
```

C.

```
free(p->ptr);
free(p);
```

D.

```
free(p);
```

```
free(p->ptr);
free(p->name);
```

4. 读下列代码：

```
char a=100;
char b=150;
unsigned char c;
c=(a<b)? a:b;
```

请问 c 的值为（B　）

A. 100　B. 150　C. -106 D. 204

5. 有下列代码：

```
int main(void)
{
  int i=1,j=2,k=3;
  if(i++==1&&(++j==3||k++==3))
k+=2;
  printf(“%d %d %d\n”,i,j,k);
}
```

程序的运行结果为（ C ）

A. 1 2 3　B. 2 3 4　C. 2 3 5　D. 2 3 6

6. 下列函数声明中有语法错误的有：（ A　）

A.　int func (float a[][] , int n);

B.　int func (float *a[] ,int n);

C.　int func (float a[] ,int n);

D.　int func (float (*a)[] ,int n);

7. 下列选项中可以作为 C 语言合法常量的有（ A）

A. -80.

B. -080

C. -8e1.0

D. -80.0e

8. 变量定义如下：

```
char s[10], *p;
char c ;
p = &c  ;
```

则下列语句不正确的有（CD　）

A. p=s+4;

B. *p=s[0];

C. s=”sist”;

D. s=p+1;

9. 以下序列中不符合堆定义的是____D____

（102，87，100，79，82，62，84，42，22，12，68）

（102，100，87，84，82，79，68，62，42，22，12）

（12，22，42，62，68，79，82，84，87，100，102）

（102，87，42，79，82，62，68，100，84，12，22）

10. 一个具有 767 个结点的完全二叉树，其叶子结点个数为__B_____

A. 383　B. 384　C. 385　D. 386

11. 若一个具有 n 个结点、k 条边的非连通无向图是一个森林（n>k）,则该森林中必有___C__ 棵树。
A. k　　B. n　　C. n—k　　D. n+k

12.将两个长度为 n 的递增有序表归并成一个长度为 2n 的递增有序表，最少需要进行关键字比较__A____次。
A. 1　　B. n—1　　C. n　　D. 2n

13. 对 n 个元素进行快速排序时，最坏情况下的时间复杂读为___D____
A. O (log2n) B. O (n)　C. O (nlog2n)　D. O (n2)

1　C 语言命令参数的表达形式是 (D)
A) main(int argc,int argv)　B)　main(int argc,char argv[])
C) main(int argc,char*argv)　D)　main(int argc,char*argv[])

( )　2　在 C 语言中，要求运算数必须是整形的运算符是 D
A) /　　B) ++　C) !=　　D) %

( )　3　在 C 语言中，数组名作为参数传递给函数，作为实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

（）　4　设有如下：　A
int *ptr[];
则一下叙述中正确的是
ptr是指向一维数组的指针变量
ptr是指向 int型数据的指针变量
ptr是指向函数的指针，该函数返回一个 int型数据
ptr是一个函数名，该函数的返回值是指向 int形数据的指针

（）5　若以下变量均是整形，且 num=sum=7;　则执行表达式　B
sum=num++;
sum++;
++num　后 sum　的值为
A)　7　　B) 8　　C) 9　D) 10

(?)　6 设整数 a 用两个字节表示。表达式 a&0x7FFF　的作用是:（0111111111111111　）C
A)　置 a 的最低位为 0;　　B) 置 a 的最低位为 1;
C)　置 a 的最高位为 0;　　D) 置 a 的最高位为 1;

（）7　类型定义：(字符串自动加\0)
char S[3]=“AB”;
char *p;
在执行了语句 P=S　之后，*(P+2)的值是（C ）。
A) „B　　B)　　„\0　　C ）不确定　　D 字符　B的地址

（）8 设有如下定义：
struct sk{int a;float b;}data,*p;
若有 p=&data;,则对 data 中的 a 的域的正确引用是
A)(*P).data.a　　B) (*P).a
C) P->data.a　　D)p.data.a

( )　9 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为“假”。则以下不能满足要求的表达式：C

A) A%2 = =1 B) !(A%2 = = 0) C) !(A%2) D) A%2

( ) 10 下面程序的功能是将字符串 s 中所有的字符 c 的删除，请选择填空

```
#include <stdio.h>
main(){
char  s[80];
int I,j;
gets(s);
for(i=j=0;s[i]!='\0';i++)
if(s[i]!=  c  )
------------------------
puts(s);
}
```

A) s[j]=s[i];j++; B)s[++j]=s[i];

C) s[j++]=s[i]; D)s[j]=s[i];

( )11请阅读以下程序:

```
#include<stdio.h>
#include<stdio.h>
main(){
int a=5,b=0,c=0;
if(a=b+c)printf("***");
else printf("$$$");
}
```

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) 输入*** D) 输出$$$

( ) 12 有以下程序

```
#include <stdio.h>
int fuc(int a){
  a=100;
return(a);
}
main( )
{
int a=20;
a=a+fuc(a);
printf("%d",a);
}
```

输出结果是:

A) 20 B) 120 C) 40 D) 200

( )13 以下程序的运行结果是:

```
  main() {
 int m=5;
 if(m++>5) printf("%d",m);
else  printf("%d",m-);
```

}

A) 4　　B)　5　　C)　6　　D)　7

( ) 14执行以下程序段后

```
int a,b,c;
a=84;
b=59;
c=a&&b; （逻辑与，&是按位与）
```

变量 c 的值是:

A)　16　　B) 8　　C) 1　　D)　0

(?) 15下面程序的运行结果是

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

A) mogninr　　B)　mo　　C)　morning　　D)　morning

( ) 16　有以下程序

```
#include <stdio.h>
 struct stu
 {int num;
   char name[10];
   int age;

};
void fun(struct stu *p)
{
printf(“%s”,(*p).name);
}
main()
{
struct stu student[3]={{9801,”Zhang”,20},
                       {9802,”Wang”,19},
                       {9803,”Zhao”,18}};
fun(student+2); }
```

输出结果是

Zhang　　B)Zhao　　C)　Wang　　D) 18

1.　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 B

A)k=*p1+*p2;　　B)p2=k;

C)p1=p2;　　D)k=*p1 *(*p2);

2. 设有如下定义：

```
int arr[]={6,7,8,9,10};
int *ptr;

ptr=arr;

*(ptr+2)+=2;

printf ("%d,%d\n",*ptr,*(ptr+2));
```

则程序段的输出结果为 D

A)8,10　　B)6,8　　C)7,9　　D)6,10

3. 执行以下程序段后,m 的值为 A

```
int a[2][3]={{1,2,3},{4,5,6}};
int m,*p;
p=&a[0][0];
m=(*p)*(*(p+2))*(*(p+4));
```

A)15　　B)14　　C)13　　D)12

4
定义一个指向函数的指针，该函数的返回值是一个指针

定义一个指针数组，数组元素是指向函数的指针，并且改函数的返回值是一个 int型的指针

5. 有以下程序

```
main()
{
char a[]="programming",b[]="language";
char *p1,*p2;
int i;
p1=a;p2=b;
for(i=0;i<7;i++)
if(*(p1+i)==*(p2+i))
printf%"c",*(p1+i));
}
```

输出结果是 D

A)gm　　B)rg　　C)or　　D)ga

6　指针 s 所指字符串的长度为　D

```
char *s="\\"Name\\Address\n";
```

A) 19　　B) 15　　C) 18　　D)　说明不合法

7　设有说明 int(*ptr)[m]其中的标识符 ptr是　C

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

8.　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

A) pb=&x　　B) pb=x;　　C) *pb=&x;　　D) *pb=*x

9　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：　D

```
int *p;
p=__________ malloc( sizeof(int));
```

则应填入

A) int　　B) inst *　　C) (*int)　　D) (int *)

10　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　C

A) p+1　　B) a+1　　C) a++　　D) ++p

11　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

A) *p=&a;　　scanf("%ld", p);
B) p=(long *)malloc(8)　scanf("%ld", p);
C) scanf("%ld", p=&a);
D) scanf("%ld", &a);

12　设已有定义: char *st="how are you"下列程序段中正确的是 C

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);

13　设有以下语句：　D

```
struct st {int n; struct st *next;};
   static struct st a[3]={5,&a[1],7,&a[2],9,'\0'},*p;
p=&a[0];
```

则表达式_____ 的值是 6。

p + + ->n　　B) p->n + +　　C) (*p).n + +　　D) + + p->n

1．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

2．C 语言中，运算对象必须是整型的运算符是_____A____

A) %　　B) /　　C) %和/　　D) *

3．为表示关系 x≥y≥z 正确的 C 语言表达式是_____A_____

A) (x>=y)&&(y>=z)　　B) (x>=y)AND(y>=z)

C) (x>=y>=z)　　D) (x>=y)&(y>=z)

4. 若定义 x 和 y 为 double 类型，则表达式：x=2,y=x+5/2 的值是____ C_____

A) 4　　B) 4.5　　C) 4.0　　D) 3.0

5. 若变量已正确说明为 int类型，要给 a,b,c输入数据，以下正确的输入语句是 D

A) read(a,b,c);　　B) scanf(“%d%d%d”,a,b,c);

C) scanf(“%D%D%D”,&a,%b,%c);　　D) scanf(“%d%d%d”,&a,&b,&c);

6. 下列语句中符合 C 语言语法的赋值语句是 ____C_____

A) a=7+b+c=a+7;　　B) a=7+b++=a+7;

C) a=7+b,b++,a+7　　D) a=7+b,c=a+7;

7. 设 int c=5和 int a, a=2+(c+=c++,c+8,++c)则 a=____B_____

A) 15　　B) 14　　C) 13　　D) 16

8．以下程序的输出结果是_____D__

```
main()
{  int x=10,y=10;
   printf(“%d %d”,x--,--y);
}
```

A) 10 10　　B) 9 9　　C) 9 10　　D) 10 9

9. 不合法的八进制数是___B_____

A) 0　　B) 028　　C) 077　　D)01

10. 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___B_____

A) a<>0　　B) !a　　C) a=0　　D) a!=0

11. 在以下运算符中，优先级最高的运算符是_____B____

A) ? :　　B) ++　　C) &&　　D) +=

12. 在 C 语言中，逻辑值“真”用_____D_____ 表示。

A) true　　B) 大于 0 的数　　C) 非 0 的整数　　D) 非 0 的数

13. 在 16 位机系统中，下面程序的输出是 A

```
main()
{  unsigned a=32768;
   printf(“a=%d,a);
}
```

A) a=32768　　B) a=32767　　C) a=-32768　　D) a=-1

14. 在 C 语言的 if语句中，用作判断的表达式为___D_____
A) 关系表达式　　B) 逻辑表达式　　C) 算术表达式　　D) 任意表达式

15. 执行下面的程序后，a 的值为 B

```
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;
   }
}
```

A) 7　　B) 8　　C) 9　　D) 10

16. 表达式_____D____ 是满足：当 c 的值为 1、3、5 三个数时值为 "真"，否则值为 "假"的表达式
A) (c=1)||(c=3)||(c=5)　　B) (c!=1)&&(c!=3)&&(c!=5)
C) (c==1)&&(c==3)&&(c=5)　　D) (c==1)||(c==3)||(c==5)

17. 若变量已正确说明，则以下程序段输出为_____A____

```
a=10;b=50;c=30;
if (a>b)
   a=b,b=c;
c=a;
```

A) a=10　b=50　c=10　　B) a=10　b=30　c=10
C) a=50　b=30　c=10　　D) a=50　b=30　c=50

18. 定义如下变量：B

```
int n=10;
则下列循环的输出结果是
while (n>7)
{  n--;
   printf( "%d" ,n);
}
```

A) 10　　B) 9　　C) 10　　D) 9
9　　8　　9　　8
8　　7　　8　　7
　　　　7　　6

19. 在 C 语言中，while语句中的条件为_____A___ 时，结束该循环。
A) 0　　B) 1　　C) true　　D) 非 0

20. 若 k 为整型，以下 while循环___C_____ 次。

```
int k=10;
while (k=0) k=k-1;
```

A) 10 次　　B) 无限循环

C) 一次不执行　　D) 执行一次

以下程序的运行结果为_____

```
int  x = 0,   y = 0;
void func1 (intx, int y)
{
x  =   5,  y   =   6;
}
void  func2()
{
        x   =  12,  y  =  13;
}
void main (void)
{
        Func1(x, y);
        printf(“%d,%d”,x,y);
        func2();
printf(“%d,%d”,x,y);
}
```

A, 5,6,12,13;　　B,0,0,12,13;　　C,5,6,0,0,　　D,0,0,0,0

2．请正确指出下列指针的最确切的含义，int(*ptr)[3],int*(*ptr)[4],

Void*(*ptr)(void*)_____。_

一个指针数组，数组元素的个数为 3 个；一个指向 4 个元素数组的指针，数组的每个元素是一个指针；一个函数的指针，这个函数的返回值还是个指针；

一个指向 3 个元素数组的指针；一个指向 4 个元素数组的指针，数组的每个元素是一个指针；一个函数的指针，这个函数的返回值还是个指针；

一个指针数组，数组元素的个数为 3 个；一个指针数组，数组的每个元素是一个指针；一个函数的指针，这个函数的返回值还是个指针；

一个指向 3 个元素数组的指针；一个指针数组，数组的每个元素是一个指针；一个函数的指针，这个函数的返回值还是个指针；

3．判断程序的结果：

```
Char*p = “HelloWorld”;
    *p = “A”
```

程序执行后的结果_____ 。

p 指向的字符串为“AelloWorle”;

p 指向的字符串为“Hello Worle”

不能通过编译；

结果未知；

4．请问这个结构体所占的空间大小是_____ 。

```
     Typedef   struct
{
    Int  a,
Char   b,
```

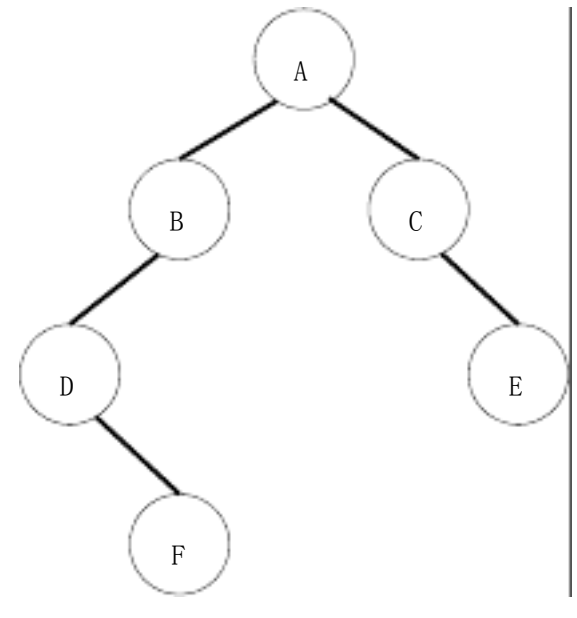
Short c,
Short d,
}AA_t;
A．16 byte; B. 9 byte;C.12 byte; D.8 byte;
5.下列排序算法中____个算法可能会出现下面情况，初始数据有序时花费的时间反而最多；
A．堆排序　　B.冒号排序　　C.快速排序　　D .SHELL 排序
6.若进栈序列为 1，2，3，4，进栈过程中可以出栈，则____不可能是一个出栈序列。
A． 1，4，3，2　　B. 2, 3, 4, 1　　C. 3, 1, 4, 2　　D. 3, 4, 2, 1
7.对线形表进行二分法查找，其前提条件是_____ ？
A.线形表以顺序方式存储，并且按关键码值排好序
B. 线形表以顺序方式存储，并且按关键码值检索频率排好序
C. 线形表以链接方式存储，并且按关键码值排好序
D.线性表以链接方式存储，并且按关键码值的检索频率排好序
在C 语言中，while语句中的条件为____时，结束该循环。
A、0　　B、1　　C、 true　　D、非0
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);
设有以下宏定义：____
#define N 3
#define Y(n) ((N)*n)
则执行语句 ：z = 2*(N*(Y(n))后，z 的值为____
A、出错　　B、72　　C、24　　D、　　36
设有说明 int S[2] = {0,1}, int *p=s;则下列错误的c 语句是____
A、s+=1；　　B、p+=1；　　C、*p++;　　D、(*p) ++;
在C 语言中，若已经定义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

二、读程序并完成填空（每空1 分， 总计10 分）
1．函数long fun(char *s)的功能是：从左到右顺序取出非空字符串str中的数字字符，形成一个十进制的整数。例如：如字符串str的值为”f3gd5.j<47s6kp2”则函数的返回值应该为数值354762 。
long fun(char *str)
{
  int i=0;
  long k=0;
  char *p=str;
  while(*p!=0 && 【1】i<9)
  {
if(*p>= 0 && *p<= 9)
{
   k= k*10【2】 + *p-0 ;
   ++i;
}
  p++ 【3】 ;
  }
return k;

}

2.一棵非空二叉树中 “最左下”结点的定义为：若树根的左子树为空，则树根为 “最左下”结点；否则，从树根的左子树的根出发，沿该结点的左孩子分支向下查找，直到某个结点不存在左孩子为止，该结点即为此二叉树的 “最左下”结点。

例如右图所示的以 A 为根的二叉树的 “最左下”结点为 D，而以 C 为根的子二叉树的 “最左下”结点为 C。



二叉树的结点类型定义如下：

```
Typedef struct BSTNode
{
  int data;
  struct BSTNode *lch;
  struct BSTNode *rch;
} *BSTree;
```

函数 BSTree Find_Del(BSTree root)的功能是：若 root指向一棵二叉树的根结点，则找出该结点的右子树上的 “最左下”结点 *p，并从树下删除以*p 为根的子树，函数返回被删除的子树的根结点指针。若该树的右子树上不存在 “最左下”结点，则返回空指针。

函数如下：

```
BSTree Find_Del(BSTree root)
{
  BSTree p,pre ;
  if(!root) return NULL;
【1】pre = root  ;
 P = pre->rch ;
  if(!p) return NULL;
     【2】  p = p->lch ;
  while( 【3】 p )
  {
    pre=p;
    p=    【4】p->lch   ;
  }
  if( 【5】 pre  ==root)
     pre->rch=NULL;
  else
       【6】pre->rch = NULL   ;
  return   【7】pre   ;
}
```

1. 计算 sizeof表达式和 strlen表达式的值

```
Char   s1[ ] =”  ”  ;
Char   s2[ ] =”Hello world”  ;
Char   *p = s2 ;
Char   *q =NULL ;
Void   *r = malloc (100) ;
```

请计算：

```
sizeof (s1) = 2;
sizeof (s2) =12;
sizeof (p) =4;
```

```
sizeof (q) =4;
sizeof (r) = 4;
```

(1)下面程序的输出是 ___99___

```
#define PR(ar)printf("%d",ar)
main( )
{
int j,a[]={1,3,5,7,9,11,13,15},*p=a+5;
For(j=3;j;j--)
{
switch(j)
{
case1:
case2:PR(*p++);break;
case3:PR(*(--p));}}}
```

(2)下面程序的输出是

___12___

```
unsigned fun6(unsigned num)
{     unsigned k=1;
do {
k*=num%10;
num/=10;
}while(num);
return(k);}
main()
{   unsigned n=26;
printf("%d\n",fun6(n));
}
```

(3)下面程序的输出是 _____9_

```
long fun5(int n)
{    long s;
if((n==1)||(n==2))
s=2;
else.
s=n+fun5(n-1);
return(s);}
main()
{   long x;
x=fun5(4);
printf("%ld\n",x);
}
```

1. 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);

2. 设有如下定义：

```
int arr[]={6,7,8,9,10};
int *ptr;

ptr=arr;

*(ptr+2)+=2;

printf ("%d,%d\n",*ptr,*(ptr+2));
```

则程序段的输出结果为

A)8,10　　B)6,8　　C)7,9　　D)6,10

3.　执行以下程序段后,m 的值为

```
int a[2][3]={{1,2,3},{4,5,6}};
int m,*p;
p=&a[0][0];
p=*a;
m=(*p)*(*(p+2))*(*(p+4));
```

a 是数组指针

A)15　　B)14　　C)13　　D)12

4

定义一个指向函数的指针，该函数的返回值是一个指针

定义一个指针数组，数组元素是指向函数的指针，并且改函数的返回值是一个 int型的指针

5.　有以下程序

```
main()
{
char a[]="programming",b[]="language";
char *p1,*p2;
int i;
p1=a;p2=b;
for(i=0;i<7;i++)
if(*(p1+i)==*(p2+i))
printf%c",*(p1+i));
}
```

输出结果是

A)gm　　B)rg　　C)or　　D)ga

6　　指针 s 所指字符串的长度为

```
char *s="\\"Name\\Address\n";
```
A) 19　　B) 15　　C) 18　　D)　说明不合法

7　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

8.　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

9　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) inst *　　C) (*int)　　D) (int *)

10　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

11　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);

12　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);

13　设有以下语句：
```
struct st {int n; struct st *next;};
   static struct st a[3]={5,&a[1],7,&a[2],9,'\0'},*p;
p=&a[0];
```
则表达式_____ 的值是 6。
A）p + + ->n　　B) p->n + +　　C) (*p).n + +　　D) + + p->n

14Include <stdio.h> include" stdio.h"有什么 区别

15 内存中的用户区主要分哪几部分?每个部分分别存储何种类型的数据?

```
char a[30];
char *b = (char *)malloc(20 * sizeof(char));
printf("%d\n", sizeof(a));
printf("%d\n", sizeof(b));
printf("%d\n", sizeof(a[3]));
printf("%d\n", sizeof(b+3));
printf("%d\n", sizeof(*(b+4)));
```

16

```
 struct A
{
    char a ;
    int i ;
    char b ;
}
sizeof(A)=?

struct B
{

    int i ;
    char a ;
    char b ;
}
sizeof(B)=?
17 char a = 128 ;
   unsigned char b = 128
    short c = a+ b ;
   unsigned short c1 = a + (char )b
    unsigned short c2 = (unsigned char)a + b ;
printf( “%d” ,c);
printf( “%d ”,c1) ;
printf( “%d ”,c2 );
```

程序的输出结果是什么 ？

0 65280 256

18 在机器中 sizeof(int) = 2 ；

写出 +3 和 -3 在机器中 的 表示 。

000000011 1111111101

```
19 int i= 1 ;
    int j= 2 ;
    double *pD = (double*)&j ;
    *pD= 3.0 ;
printf( “%d” , i);
```

分析 输出结果 并说明原因。

1074266112 覆盖了 I的空间

冒泡法排序函数：

```
void bubble(int a[],int n)
{
int i,j,k;
for(i=1,i<n;i++)
   for(j=0;j<n-i-1;j++)
   if(a[j]>a[j+1])
   {
        k=a[j];
        a[j]=a[j+1];
        a[j+1]=k;
       }
}
```

选择法排序函数：

```
void sort(int a[],int n)
{
int i,j,k,t;
for(i=0,i<n-1;i++)
   {
      k=i;
      for(j=i+1;j<n;j++)
          if(a[k]<a[j]) k=j;
       if(k!=i)
          {
          t=a[i];
          a[i]=a[k];
          a[k]=t;
          }
   }
}
```

折半查找函数（原数组有序）：

```
void search(int a[],int n,int x)
{
    int left=0,right=n-1,mid,flag=0;
    while((flag==0)&&(left<=right))
    {
        mid=(left+right)/2;
        if(x==a[mid])
         {
              printf("%d%d",x,mid);
              flag =1;
         }
```

```
        else if(x<a[mid]) right=mid-1;
                else left=mid+1;
    }
```

4. static有什么用途？（请至少说明两种）

1.限制变量的作用域

2.设置变量的存储域

7. 引用与指针有什么区别？

1) 引用必须初始化，而指针不必。

2) 引用初始化以后不能被改变，而指针可以改变指向对象。

3) 不存在指向空值的引用，却存在指向空值的指针。

8. 描述实时系统的基本特性

在特定时间内完成特定任务，实时性和可靠性

9. 全局变量和局部变量在内存中是否有区别？如果有，是什么区别？

有，全局变量储存在静态数据库中，局部变量储存在堆栈中

10. 什么是平衡二叉树？

左右子数都是平衡二叉树 且左右子数的深度差不超过 1

11. 堆栈溢出一般是由什么原因导致的？

没有回收垃圾资源

12. 什么函数不能声明为虚函数？

constructor

13. 冒泡排序算法的时间复杂度是什么？

O(n^2)

14. 写出 float x与 “零值”比较的 if语句。

if(x>0.000001&&x<-0.000001)

16. Interne采用哪种网络协议？该协议的主要层次结构？

tcp/ip应用层/传输层/网络层/数据链路层/物理层

17. Interne物理地址和 IP 地址转换采用什么协议？

ARP (Address Resolution Protocol)(地址解析協議)

18.IP地址的编码分为哪俩部分？

IP 地址由两部分组成 网络号和主机号 但必须与子网掩码按位与上之后才会区分网络位和主机位

2.用户输入 M,N 值，从 1 至 N 开始顺序循环数数，每数到 M 输出该数值，直至全部输出。写出 C 程序。

循环链表，用取余操作做

3.不能做 switch()的参数类型是：

switch的参数不能为实型。

1、局部变量能否和全局变量重名？

答：能，局部会屏蔽全局。要用全局变量，需要使用"::"

局部变量可以与全局变量同名，在函数内引用这个变量时，会用到同名的局部变量，而不会用到全局变量。对于有些编译器而言，在同一个函数内可以定义多个同名的局部变量，比如在两个循环体内都定义一个同名的局部变量，而那个局部变量的作用域就在那个循环体内

2、如何引用一个已经定义过的全局变量？

答：extern

可以用引用头文件的方式，也可以用 extern关键字，如果用引用头文件方式来引用某个在头文件中声明的全局变理，假定你将那个变量写错了，那么在编译期间会报错，如果你用 extern方式引用时，假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错

3、全局变量可不可以定义在可被多个.C 文件包含的头文件中？为什么？

答：可以，在不同的 C 文件中以 stati形式来声明同名全局变量。
可以在不同的 C 文件中声明同名的全局变量，前提是其中只能有一个 C 文件中对此变量赋初值，此时连接不会出错
4、语句 for(；1 ；)有什么问题？它是什么意思？
答：和 while(1相同。
5、do……while 和 while……do 有什么区别？
答：先作一个循环一遍再判断，先作一个判断以后再循环
6、请写出下列代码的输出内容

```
#include<stdio.h>
main()
{
int a,b,c,d;
a=10;
b=a++;
c=++a;
d=10*a++;
printf(″bc，d：%d ，%d ，%d″，b，c，d）；
return 0;
}
```

答：10，12，120
1、static全局变量与普通的全局变量有什么区别？stati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？
全局变量(外部变量)的说明之前再冠以 static就构成了静态的全局变量。全局变量本身就是静态存储方式，静态全局变量当然也是静态存储方式。 这两者在存储方式上并无不同。这两者的区别虽在于非静态全局变量的作用域是整个源程序， 当一个源程序由多个源文件组成时，非静态的全局变量在各个源文件中都是有效的。 而静态全局变量则限制了其作用域， 即只在定义该变量的源文件内有效， 在同一源程序的其它源文件中不能使用它。由于静态全局变量的作用域局限于一个源文件内，只能为该源文件内的函数公用， 因此可以避免在其它源文件中引起错误。
从以上分析可以看出， 把局部变量改变为静态变量后是改变了它的存储方式即改变了它的生存期。把全局变量改变为静态变量后是改变了它的作用域， 限制了它的使用范围。
static函数与普通函数作用域不同。仅在本文件。只在当前源文件中使用的函数应该说明为内部函数(static)内部函数应该在当前源文件中说明和定义。对于可在当前源文件以外使用的函数，应该在一个头文件中说明，要使用这些函数的源文件要包含这个头文件
static全局变量与普通的全局变量有什么区别：static全局变量只初使化一次，防止在其他文件单元中被引用;
stati局部变量和普通局部变量有什么区别：stati局部变量只被初始化一次，下一次依据上一次结果值;
static函数与普通函数有什么区别：static函数在内存中只有一份，普通函数在每个被调用中维持一份拷贝
2、程序的局部变量存在于（堆栈）中，全局变量存在于（静态区 ）中，动态申请数据存在于（ 堆）中。
3、设有以下说明和定义：

```
typedef union {long i; int k[5]; char c;} DATE;                    20
struct data { int cat; DATE cow; double dog;} too;                   4+20+8=32
DATE max;
```

则语句 printf("%d",sizeof(struct date)+sizeof(max));的执行结果是：（52）
答：DATE 是一个 union, 变量公用空间. 里面最大的变量类型是 int[5]占用 20 个字节. 所以它的大小是 20
data 是一个 struct,每个变量分开占用空间. 依次为 int4 + DATE20 + double8 = 32.

所以结果是 20 + 32 = 52.
当然在某些 16 位编辑器下, int可能是 2 字节,那么结果是 int2 + DATE10 + double8 = 20
4、队列和栈有什么区别?
队列先进先出，栈后进先出
5、写出下列代码的输出内容

```
#include<stdio.h>
int inc(int a)
{
return(++a);
}
int multi(int*a,int*b,int*c)
{
return(*c=*a**b);
}
typedef int(FUNC1)(int in);
typedef int(FUNC2) (int*,int*,int*);
void show(FUNC2 fun,int arg1, int*arg2)
{
INCp=&inc;
int temp =p(arg1);
fun(&temp,&arg1, arg2);
printf("%d\n",*arg2);
}
main()
{
int a;
show(multi,10,&a);
return 0;
}
```

答：110
7、请找出下面代码中的所以错误
说明：以下代码是把一个字符串倒序，如“abcd倒序后变为“dcba”

```
1、#include"string.h"
2、main()
3、{
4、 char*src="hello,world";
5、 char* dest=NULL;
6、 int len=strlen(src);
7、 dest=(char*)malloc(len);
8、 char* d=dest;
9、 char* s=src[len];
10、 while(len--!=0)
11、 d++=s--;
12、 printf("%s",dest);
13、 return 0;
```

14、}
答:
方法 1:
```
int main(){
char* src = "hello,world";
int len = strlen(src);
char* dest = (char*)malloc(len+1);要为\0分配一个空间
char* d = dest;
char* s = &src[len-1];指向最后一个字符
while( len-- != 0 )
*d++=*s--;
*d = 0;尾部要加\0
printf("%s\n",dest);
free(dest);使用完，应当释放空间，以免造成内存汇泄露
return 0;
}
```
方法 2:
```
#include <stdio.h>
#include <string.h>
main()
{
char str[]="hello,world";
int len=strlen(str);
char t;
for(int i=0; i<len/2; i++)
{
t=str[i];
str[i]=str[len-i-1]; str[len-i-1]=t;
}
printf("%s",str);
return 0;
}
```
1.-1,2,7,28,,126 请问 28 和 126 中间那个数是什么？为什么？
第一题的答案应该是 4^3-1=63
规律是 n^3-1(当 n 为偶数 0，2，4)
n^3+1(当 n 为奇数 1，3，5)
答案：63
2.用两个栈实现一个队列的功能？要求给出算法和思路！
设 2 个栈为 A,B, 一开始均为空.

入队:
将新元素 push 入栈 A;

出队:
(1)判断栈 B 是否为空;
(2)如果不为空，则将栈 A 中所有元素依次 pop 出并 push 到栈 B;

(3)将栈 B 的栈顶元素 pop 出；

这样实现的队列入队和出队的平摊复杂度都还是 O(1)，比上面的几种方法要好。3.在 c 语言库函数中将一个字符转换成整型的函数是 atool()吗，这个函数的原型是什么？
函数名: atol
功 能:把字符串转换成长整型数
用 法: long atol(const char *nptr);
程序例:

```
#include <stdlib.h>
#include <stdio.h>
int main(void)
{
long l;
char *str = "98765432";
l = atol(lstr);
printf("string = %s integer = %ld\n", str, l);
return(0);
}
```

2.对于一个频繁使用的短小函数,在 C 语言中应用什么实现,在 C++ 中应用什么实现?
c 用宏定义，c++用 inline
3.直接链接两个信令点的一组链路称作什么?
PPP 点到点连接
4.接入网用的是什么接口?
5.voip都用了那些协议?
6.软件测试都有那些种类?
黑盒：针对系统功能的测试 白合：测试函数功能，各函数接口
7.确定模块的功能和模块的接口是在软件设计的那个队段完成的?
概要设计阶段
1.IP Phone 的原理是什么?
IPV6
2.TCP/IP 通信建立的过程怎样，端口有什么作用？
三次握手，确定是哪个应用程序使用该协议
3.1 号信令和 7 号信令有什么区别，我国某前广泛使用的是那一种？
4.列举 5 种以上的电话新业务？

微软亚洲技术中心的面试题!!!
1．进程和线程的差别。
线程是指进程内的一个执行单元,也是进程内的可调度实体.
与进程的区别:
(1)调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位
(2)并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可并发执行
(3)拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源.

(4)系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销。
2.测试方法
人工测试：个人复查、抽查和会审
机器测试：黑盒测试和白盒测试

2．Heap 与 stack的差别。
Heap 是堆，stack是栈。
Stack 的空间由操作系统自动分配/释放，Heap 上的空间手动分配/释放。
Stack 空间有限，Heap 是很大的自由存储区
C 中的 malloc 函数分配的内存空间即在堆上,C++中对应的是 new 操作符。
程序在编译期对变量和函数分配内存都在栈上进行,且程序运行过程中函数调用时参数的传递也在栈上进行
3．Windows 下的内存是如何管理的?
4．介绍.Net和.Net的安全性。
5．客户端如何访问.Net组件实现 Web Service？
6．C/C++ 编译器中虚表是如何完成的?
7．谈谈 COM 的线程模型。然后讨论进程内/外组件的差别。
8．谈谈 IA32 下的分页机制
小页(4K)两级分页模式，大页(4M)一级
9．给两个变量，如何找出一个带环单链表中是什么地方出现环的?
一个递增一，一个递增二，他们指向同一个接点时就是环出现的地方
10．在 IA32 中一共有多少种办法从用户态跳到内核态?
通过调用门，从 ring3到 ring0，中断从 ring3到 ring0，进入 vm86 等等
11．如果只想让程序有一个实例运行，不能运行两个。像 winamp 一样，只能开一个窗口，怎样实现?
用内存映射或全局原子（互斥变量）、查找窗口句柄..
FindWindow ，互斥，写标志到文件或注册表,共享内存。.
12．如何截取键盘的响应，让所有的„a变成„b?
键盘钩子 SetWindowsHookEx
13．Apartment 在 COM 中有什么用？为什么要引入?
14．存储过程是什么？有什么用？有什么优点?
我的理解就是一堆 sql的集合，可以建立非常复杂的查询，编译运行，所以运行一次后，以后再运行速度比单独执行 SQL 快很多
1. 用宏定义写出 swap （x，y）

```
#define swap(x, y)\
x = x + y;\
y = x - y;\
x = x - y;
```

2.数组 a[N]，存放了 1 至 N-1 个数，其中某个数重复一次。写一个函数，找出被重复的数字.时间复杂度必须为 o（N）函数原型：

```
int do_dup(int a[],int N)
```

3 一语句实现 x 是否为 2 的若干次幂的判断

```
int i = 512;
cout << boolalpha << ((i & (i - 1)) ? false : true) << endl;
```

4.unsigned int intvert(unsigned int x,int p,int n)实现对 x 的进行转换,p为起始转化位,n为需要转换的长度,假设起始点在右边.如 x=0b0001 0001,p=4,n=3 转换后 x=0b0110 0001

```
unsigned int intvert(unsigned int x,int p,int n){
unsigned int _t = 0;
unsigned int _a = 1;
for(int i = 0; i < n; ++i){
_t |= _a;
_a = _a << 1;
}
_t = _t << p;
x ^= _t;
return x;
}
```

什么是预编译

何时需要预编译：

１、总是使用不经常改动的大型代码体。

２、程序由多个模块组成，所有模块都使用一组标准的包含文件和相同的编译选项。在这种情况下，可以将所有包含文件预编译为一个预编译头。

```
char * const p;
char const * p
const char *p
```

上述三个有什么区别？

char * const p;常量指针，p 的值不可以修改

char const *;p;指向常量的指针，指向的常量值不可以改

const char *p ;和 char const *p

```
char str1[] = "abc";
char str2[] = "abc";

const char str3[] = "abc";
const char str4[] = "abc";

const char *str5 = "abc";
const char *str6 = "abc";

char *str7 = "abc";
char *str8 = "abc";


cout << ( str1 == str2 ) << endl;
cout << ( str3 == str4 ) << endl;
cout << ( str5 == str6 ) << endl;
cout << ( str7 == str8 ) << endl;
```

结果是：0 0 1 1

解答：str1,str2,str3,str4是数组变量，它们有各自的内存空间；

而 str5,str6,str7,str8是指针，它们指向相同的常量区域。

12. 以下代码中的两个 sizeof用法有问题吗？[C 易]

```
void UpperCase( char str[] 将/str中的小写字母转换成大写字母
{
    for( size_t i=0; i<sizeof(str)/sizeof(str[0]); ++i )
        if( 'a'<=str[i] && str[i]<='z' )
            str[i] -= ('a'-'A' );
}
char str[] = "aBcDe";
cout << "st字符长度为: " << sizeof(str)/sizeof(str[0]) << endl;
UpperCase( str );
cout << str << endl;
```

答：函数内的 sizeof有问题。根据语法，sizeof如用于数组，只能测出静态数组的大小，无法检测动态分配的或外部数组大小。函数外的 str是一个静态定义的数组，因此其大小为 6，函数内的 str实际只是一个指向字符串的指针，没有任何额外的与数组相关的信息，因此 sizeof作用于上只将其当指针看，一个指针为 4 个字节，因此返回 4。

一个 32 位的机器,该机器的指针是多少位

指针是多少位只要看地址总线的位数就行了。80386 以后的机子都是 32 的数据总线。所以指针的位数就是 4 个字节了。

```
main()
{
  int a[5]={1,2,3,4,5};
   int *ptr=(int *)(&a+1);

   printf("%d,%d",*(a+1),*(ptr-1));
}
```

输出：2,5

(a+1) 就是 a[1]，(ptr-1)就是 a[4]执行结果是 2，5

&a+1 不是首地址+1，系统会认为加一个 a 数组的偏移，是偏移了一个数组的大小（本例是 5 个 int）

int *ptr=(int *)(&a+1);

则 ptr实际是&(a[5])也就是 a+5

原因如下：

&a 是数组指针，其类型为 int (*)[5];

而指针加 1 要根据指针类型加上一定的值，

不同类型的指针+1 之后增加的大小不同

a 是长度为 5 的 int数组指针，所以要加 5*sizeof(int)

所以 ptr实际是 a[5]

但是 prt与(&a+1)类型是不一样的(这点很重要)

所以 prt-1只会减去 sizeof(int*)

a,&a 的地址是一样的，但意思不一样，a 是数组首地址，也就是 a[0]的地址，&a 是对象（数组）首地址，a+1 是数组下一元素的地址，即 a[1],&a+1是下一个对象的地址，即 a[5].

1.请问以下代码有什么问题：

```
int main()
{
char a;
char *str=&a;
strcpy(str,"hello");
printf(str);
return 0;
}
```

没有为 str分配内存空间，将会发生异常

问题出在将一个字符串复制进一个字符变量指针所指地址。虽然可以正确输出结果，但因为越界进行内在读写而导致程序崩溃。

```
char* s="AAA";
printf("%s",s);
s[0]='B';
printf("%s",s);
```

有什么错？

"AAA"是字符串常量。s 是指针，指向这个字符串常量，所以声明 s 的时候就有问题。

```
cosnt char* s="AAA";
```

然后又因为是常量，所以对是 s[0]的赋值操作是不合法的。

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 Min(X, Y) ((X)>(Y)?(Y):(X))//结尾没有;
```

2、嵌入式系统中经常要用到无限循环，你怎么用 C 编写死循环。

```
while(1)或者 for(;;)
```

3、关键字 static的作用是什么？

定义静态变量

4、关键字 const 有什么含意？

表示常量不可以修改的变量。

5、关键字 volatile有什么含意？并举出三个不同的例子？

提示编译器对象的值可能在编译器未监测到的情况下改变。

```
int (*s[10])(int)表示的是什么啊
int (*s[10])(int)函数指针数组，每个指针指向一个 int func(int param)的函数。
```

1.有以下表达式：

```
int a=248; b=4;int const c=21;const int *d=&a;
int *const e=&b;int const *f const =&a;
```

请问下列表达式哪些会被编译器禁止？为什么？

```
*c=32;d=&b;*d=43;e=34;e=&a;f=0x321f;
```

*c 这是个什么东东，禁止
*d 说了是 const， 禁止
e = &a 说了是 const 禁止
const *f const =&a禁止
2.交换两个变量的值，不使用第三个变量。即 a=3,b=5,交换之后 a=5,b=3;
有两种解法, 一种用算术算法, 一种用^(异或)
a = a + b;
b = a - b;
a = a - b;
or
a = a^b;/只能对 int,char..
b = a^b;
a = a^b;
or
a ^= b ^= a;
3.c和 c++中的 struc有什么不同?
c 和 c++中 struc的主要区别是 c 中的 struc不可以含有成员函数，而 c++中的 struc可以。c++中 struct 和 class的主要区别在于默认的存取权限不同，struc默认为 public，而 class默认为 private
4.#include <stdio.h>
#include <stdlib.h>
void getmemory(char *p)
{
p=(char *) malloc(100);
strcpy(p,"hello world");
}
int main( )
{
char *str=NULL;
getmemory(str);
printf("%s/n",str);
free(str);
return 0;
}
程序崩溃，getmemory 中的 malloc 不能返回动态内存， free( ) 对 str操作很危险
5.char szstr[10];
strcpy(szstr,"0123456789");
产生什么结果? 为什么?
长度不一样，会造成非法的 OS
6.列举几种进程的同步机制，并比较其优缺点。
原子操作
信号量机制
自旋锁
管程，会合，分布式系统

7.进程之间通信的途径
共享存储系统

消息传递系统
管道：以文件系统为基础
11.进程死锁的原因
资源竞争及进程推进顺序非法
12.死锁的 4 个必要条件
互斥、请求保持、不可剥夺、环路
13.死锁的处理
鸵鸟策略、预防策略、避免策略、检测与解除死锁
15.　操作系统中进程调度策略有哪几种？
FCFS(先来先服务)，优先级，时间片轮转，多级反馈
8.类的静态成员和非静态成员有何区别？
类的静态成员每个类只有一个，非静态成员每个对象一个
9.纯虚函数如何定义？使用时应注意什么？
virtual void f()=0;
是接口，子类必须要实现
10.数组和链表的区别
数组：数据顺序存储，固定大小
连表：数据可以随机存储，大小可动态改变

12.ISO 的七层模型是什么？tcp/udp是属于哪一层？tcp/udp有何优缺点？
应用层
表示层
会话层
运输层
网络层
物理链路层
物理层
tcp /udp属于运输层
TCP　服务提供了数据流传输、可靠性、有效流控制、全双工操作和多路复用技术等。
与 TCP　不同， UDP　并不提供对 IP 协议的可靠机制、流控制以及错误恢复功能等。由于 UDP　比较简单， UDP　头包含很少的字节，比 TCP　负载消耗少。
tcp:提供稳定的传输服务，有流量控制，缺点是包头大，冗余性不好
udp: 不提供稳定的服务，包头小，开销小

1：(void *)ptr和 (*(void**))ptr的结果是否相同？其中 ptr为同一个指针
.(void *)ptr和 (*(void**))ptr值是相同的
2：int main()
　{
　 int x=3;
　 printf("%d",x);
　 return 1;

　}
问函数既然不会被其它函数调用，为什么要返回 1？
mian 中，c 标准认为 0 表示成功，非 0 表示错误。具体的值是某中具体出错信息

1，要对绝对地址 0x100000 赋值，我们可以用
(unsigned int*)0x100000 = 1234;
那么要是想让程序跳转到绝对地址是 0x100000 去执行，应该怎么做？
*((void (*)( ))0x100000) ( );
首先要将 0x100000 强制转换成函数指针,即:
(void (*)())0x100000
然后再调用它:
*((void (*)())0x100000)();
用 typedef可以看得更直观些:
typedef void(*)() voidFuncPtr;
*((voidFuncPtr)0x100000)();
2，已知一个数组 table，用一个宏定义，求出数据的元素个数
#define NTBL
#define NTBL (sizeof(table)/sizeof(table[0]))

面试题: 线程与进程的区别和联系？线程是否具有相同的堆栈？dl是否有独立的堆栈?
进程是死的，只是一些资源的集合，真正的程序执行都是线程来完成的，程序启动的时候操作系统就帮你创建了一个主线程。

每个线程有自己的堆栈。
DLL 中有没有独立的堆栈，这个问题不好回答，或者说这个问题本身是否有问题。因为 DLL 中的代码是被某些线程所执行，只有线程拥有堆栈，如果 DLL 中的代码是 EXE 中的线程所调用，那么这个时候是不是说这个 DLL 没有自己独立的堆栈？如果 DLL 中的代码是由 DLL 自己创建的线程所执行，那么是不是说 DLL 有独立的堆栈？

以上讲的是堆栈，如果对于堆来说，每个 DLL 有自己的堆，所以如果是从 DLL 中动态分配的内存，最好是从 DLL 中删除，如果你从 DLL 中分配内存，然后在 EXE 中，或者另外一个 DLL 中删除，很有可能导致程序崩溃

```
unsigned short A = 10;
printf("~A = %u\n", ~A);

char c=128;
printf("c=%d\n",c);
```

输出多少？并分析过程
第一题，～A ＝0xfffffff5,值n为－11，但输出的是 uint，所以输出 4294967285
第二题，c＝0x10,输出的是 int，最高位为 1，是负数，所以它的值就是 0x00 的补码就是 128，所以输出－128。
这两道题都是在考察二进制向 int或 uint转换时的最高位处理。

分析下面的程序：
```
void GetMemory(char **p,int num)
```

```
{
    *p=(char *)malloc(num);

}
int main()
{
    char *str=NULL;

    GetMemory(&str,100);

    strcpy(str,"hello");

    free(str);

    if(str!=NULL)
    {
        strcpy(str,"world");
    }

    printf("\n str is %s",str);
    getchar();
}
```

问输出结果是什么？希望大家能说说原因，先谢谢了

输出 str is world

free 只是释放的 str指向的内存空间,它本身的值还是存在的.

所以 free之后，有一个好的习惯就是将 str=NULL.

此时 str指向空间的内存已被回收,如果输出语句之前还存在分配空间的操作的话,这段存储空间是可能被重新分配给其他变量的,

尽管这段程序确实是存在大大的问题（上面各位已经说得很清楚了），但是通常会打印出 world来。

这是因为，进程中的内存管理一般不是由操作系统完成的，而是由库函数自己完成的。

当你 malloc一块内存的时候，管理库向操作系统申请一块空间（可能会比你申请的大一些），然后在这块空间中记录一些管理信息（一般是在你申请的内存前面一点），并将可用内存的地址返回。但是释放内存的时候，管理库通常都不会将内存还给操作系统，因此你是可以继续访问这块地址的，只不过。。。。。。。

楼上都说过了，最好别这么干。

char a[10],strlen(a)为什么等于 15？运行的结果

```
#include "stdio.h"
#include "string.h"

void main()
{

char aa[10];
printf("%d",strlen(aa));
}
```

sizeof(和初不初始化，没有关系；
strlen(和初始化有关。

char (*str)[20];/*是str个数组指针，即指向数组的指针．*/
char *str[20];/*是str个指针数组，其元素为指针型数据．*/

long a=0x801010;
a+5=?
0x801010 用二进制表示为：“1000 0000 0001 0000 0001 0000”十进制的值为 8392720 ，再加上 5 就是 8392725 罗

1)给定结构 struct A
{
	char t:4;
	char k:4;
	unsigned short i:8;
	unsigned long m;
}问 sizeof(A) = ?
给定结构 struct A
{
	char t:4;位
	char k:4; 位
	unsigned short i:8;位
	unsigned long m; /偏移 2 字节保证 4 字节对齐
}; /共 8 字节
2)下面的函数实现在一个数上加一个数，有什么错误？请改正。
int add_n (int n)
{
    static int i = 100;
    i += n;
    return i;
}
当你第二次调用时得不到正确的结果，难道你写个函数就是为了调用一次？问题就出在 static上？

//帮忙分析一下
#include<iostream.h>
#include <string.h>
#include <malloc.h>
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
typedef structAA

```
{
        int b1:5;
        int b2:2;
}AA;
void main()
{
        AA aa;
        char cc[100];
         strcpy(cc,"0123456789abcdefghijklmnopqrstuvwxyz");
       memcpy(&aa,cc,sizeof(AA));
        cout << aa.b1 <<endl;
        cout << aa.b2 <<endl;
}
```

答案是 -16 和 1
首先 sizeof(AA)的大小为 4,b1 和 b2 分别占 5bit和 2bit.
经过 strcpy和 memcpy　后,aa 的 4 个字节所存放的值是:
0,1,2,3的 ASC　码，即 00110000,00110001,00110010,00110011
所以，最后一步：显示的是这 4 个字节的前 5 位，和之后的 2 位
分别为：10000,和 01
因为 int是有正负之分　　所以：答案是-16 和 1

求函数返回值，输入 x=9999;

```
int func( x )
{
    int countx = 0;
    while ( x )
    {
        countx ++;
        x = x&(x-1);
    }
    return countx;
}
```

结果呢?
知道了这是统计 9999 的二进制数值中有多少个 1 的函数，且有
9999 ＝9×1024 ＋512 ＋256 ＋15

9×1024 中含有 1 的个数为 2；
512 中含有 1 的个数为 1；
256 中含有 1 的个数为 1；
15 中含有 1 的个数为 4；
故共有 1 的个数为 8，结果为 8。
1000 - 1 = 0111 正好是原数取反。这就是原理。
用这种方法来求 1 的个数是很效率很高的。
不必去一个一个地移位。循环次数最少。

int a,b,请写函数实现 C=a+b ,不可以改变数据类型,如将 c 改为 long in关键是如何处理溢出问题