**C语言程序设计10**

窗体顶端

窗体底端

**一、判断题 (20分)**

1. 语句char \*p = &quot;Turbo C&quot;的含义是将字符串赋给字符型指针变量p。

http://code.hdu.edu.cn/resources/images/cross.png 2分

2. char \*p="girl";的含义是定义字符型指针变量p，p的值是字符串"girl"。

http://code.hdu.edu.cn/resources/images/cross.png 2分

3. 若指针变量指向一维数组的第一个元素，将指针变量加整数3，即指针后移3个元素。

http://code.hdu.edu.cn/resources/images/tick.png 2分

4. b是一维数组名，数组元素b[1]还可写成\*(b++)。

http://code.hdu.edu.cn/resources/images/cross.png 2分

5. 用字符指针指向一个字符串，就是使字符指针指向字符串的第一个字符

http://code.hdu.edu.cn/resources/images/tick.png 2分

6. 设有说明int　\*ptr[10];　其中ptr是指向含有10个整型元素的一维数组的指针。

http://code.hdu.edu.cn/resources/images/cross.png 2分

7. 由于指针的数据类型实际上是指定指针所能指向对象的数据类型，所以不同数据类型的指针变量不能相互赋值。

http://code.hdu.edu.cn/resources/images/tick.png 2分

8. 在C语言中,程序main(){int　\*ptr;　\*ptr=100;　printf(&quot;%d&quot;,\*ptr);}是正确的。

http://code.hdu.edu.cn/resources/images/cross.png 2分

9. 指针变量不能作为函数的参数。

http://code.hdu.edu.cn/resources/images/cross.png 2分

10. 相同类型指针的集合称为指针数组，即指针数组中的每个元素都相当于一个指针变量。

http://code.hdu.edu.cn/resources/images/tick.png 2分

**二、选择题 (22分)**

1.若有语句：char \*line[5];，以下叙述中正确的是( )。 [A] 2分

A. 定义line是一个数组，每个数组元素是一个基类型为char为指针变量。  
B. 定义line是一个指针变量，该变量可以指向一个长度为5的字符型数组。  
C. 定义line是一个指针数组，语句中的\*号称为间址运算符。  
D. 定义line是一个指向字符型函数的指针。

2.下面程序段的运行结果是( )。

char str[]="ABC",\*p=str;

printf("%d\n",\*(p+3)); [B] 2分

A. 67     B. 0     C. 字符′C′的地址     D. 字符′C′

3.char a[]="lanuage",\*p;p=a;while(\*p!=′u′){printf（"%c",\*p-32);p++;}[C] 2分

A. LANGUAGE     B. language

C. LAN     D. langUAGE

4.阅读下列程序，则执行后的结果为( )。

#include "stdio.h"

int main()

{ int c[][4]={1,2,3,4,5,6,7,34,213,56,62,3,23,12,34,56};

printf("%x,%x",c[2][2],\*(\*(c+1)+1));

} [B] 2分

A. 62,5     B. 3e,6     C. 56,5     D. 3E,6

5.下面程序段的运行结果是( )。

char \*s="abcde";

s+=2;

printf("%d",s); [C] 2分

A. cde     B. 字符′c′     C. 字符′c′的地址     D. 无确定的输出结果

6.char \*s[ ]={\"first\",\"second\",\"third\"};

printf(\"%s,%s,%c\\n\", s[0], s[1]+1,\*s[2]);[D] 2分

A. first,second,\*third     B. first, econd+1,\*third

C. first, econd, \*t     D. first, econd, t

7.若有定义“int (\*ptr)(float);”，其正确的含义是（）。 [C] 2分

A. ptr是指向一维数组的指针变量  
B. ptr是指向int型数据的指针变量  
C. ptr是指向函数的指针变量，该函数有一个float型参数，返回值是整型  
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

8.下列程序段的运行结果为（ ）。

char s[]=\"ab\\0cd\"; puts(s);[D] 2分

A. abcd     B. ab cd     C. ab0     D. ab

9.下面程序段的输出结果是（ ）。

char a[10]= \"abcd\";

char b[4]= \"ABC\";

strcat(a,b);

printf(“%s”,a); [A] 2分

A. abcdABC     B. abcd     C. ABC     D. ABCabcd

10.下面程序的输出结果是\_\_\_\_。

void main()

{

int a[12]={1,2,3,4,5,6,7,8,9,10,11,12},\*p[4],i;

for(i=0;i<4;i++)

p[i]=a+i\*3; // 或p[i]=&a[i\*3];

printf("%d", \*(p[3]+2)); // 或printf("%d", p[3][2]);

}  [C] 2分

A. 6     B. 8     C. 12     D. 程序有错

11.若有char \*a[2]={"abcd", "ABCD"};

则下面的表述正确的是（）。 [D] 2分

A. a数组元素的值分别是"abcd"和"ABCD"  
B. a是一个指针变量，它指向含有两个数组元素的字符型一维数组  
C. a数组的两个元素分别存放的是含有4个字符的一维字符数组的首地址  
D. a数组的两个元素中各自存放了字符'a'和'A'的地址

**三、程序填空题 (5分)**

1. 求二维数组a中的最大值与b中的最大值之差。 5分

#include<stdio.h>

float find\_max(\_\_(1)\_\_)

{

int i,j;

float max=\*\*x;

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

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

if(\*(\*(x+i)+j)>max)

max=\_\_(2)\_\_;

return max;

}

void main()

{

float a[5][5],b[6][4], \_\_(3)\_\_;

int i,j;

for(i=0;i<5;i++)

pa[i]=a[i];

for(i=0;i<6;i++)

\_\_(4)\_\_;

for(i=0;i<5;i++)

for(j=0;j<5;j++)

scanf("%f",&a[i][j]);

for(i=0;i<6;i++)

for(j=0;j<4;j++)

scanf("%f",&b[i][j]);

printf("%.1f\n",find\_max(pa,5,5)-find\_max(pb,6,4));

}

(1)答案 2分

float \*x[],int m,int n

(2)答案 1分

\*(\*(x+i)+j)

(3)答案 1分

\*pa[5],\*pb[6]

(4)答案 1分

pb[i]=b[i]

**四、程序设计题 (40分)**

1. 回文串 10分

题目描述

“回文串”是一个正读和反读都一样的字符串，比如“level”或者“noon”等等就是回文串。请写一个程序判断读入的字符串是否是“回文”。

输入描述

每组测试数据包含一个字符串。

输出描述

如果一个字符串是回文串，则输出"yes",否则输出"no".

样例输入

level

abcde

noon

haha

样例输出

yes

no

yes

no

#include <stdio.h>

#include <string.h>

**int** special**(char \***l**,int** len**) //本题和最后一题差不多，通过指针，详见最后一题**

**{**

**char \***p1**,\***p2**;**

p1**=**l**;**

p2**=&**l**[**len**-**1**];**

**while(**p1**<**p2**)**

**{**

**if(\***p1**==\***p2**) //若相等则继续询问~下一对~**

**{**

p1**++;**

p2**--;**

**}**

**Else //若不等，则不是回文，返回0（表示不是回文）**

**return** 0**;**

**}**

**return** 1**;**

**}**

**int main()**

**{**

**char** s**[**81**]; //为什么是81，张桦老师说过的，，，**

**while(**scanf**(**"%s"**,**s**)!=**EOF**)**

**{**

**if(**special**(**s**,**strlen**(**s**)))**

puts**(**"yes"**);**

**else**

puts**(**"no"**);**

**}**

**return** 0**;**

**}**

2. 首字母变大写 10分

题目描述

输入一个英文句子，将每个单词的第一个字母改成大写字母。

输入描述

测试实例是一个长度不超过100的英文句子。

输出描述

请输出按照要求改写后的英文句子。

样例输入

i like acm

样例输出

I Like Acm

#include <stdio.h>

#include <string.h>

**int main()**

**{**

**char** buf**[**81**]; //这个应该用buf[100];因为题目已指定**

**//这题思路不太好，可跳过**

**int** i**=**0**; //与第3题可以参考第3题的思路，进行改动**

**while(**scanf**(**"%c"**,&**buf**[**i**])!=**EOF**)**

**{**

**if(**buf**[**i**]==**' '**||**buf**[**i**]==**'\n'**)**

**{**

**if(**buf**[**0**]>=**'a'**)**

buf**[**0**]-=**32**;**

**if(**buf**[**i**]==**' '**)**

**{**

buf**[**i**]=**'\0'**;**

i**=-**1**;**

printf**(**"%s "**,**buf**);**

**}**

**else**

**{**

buf**[**i**]=**'\0'**;**

printf**(**"%s\n"**,**buf**);**

**continue;**

**}**

**}**

i**++;**

**}**

**return** 0**;**

**}**

3. 删除字符串中的空格，并统计空格个数 10分

题目描述

编写函数int strdel(char \*s)的定义，它实现将指针s所指向的字符串中的所有空格字符删除，并统计所删除的空格字符的个数返回给主函数。编写主程序，验证strdel()的有效性。

输入描述

一字符串

输出描述

字符串中的空格个数以及删除空格后的字符串

样例输入

qw e rt yu iop

样例输出

Please input a string:

字符串中的空格个数为:5

删除空格后字符串变为:

qwertyuiop

#include<stdio.h>

**int** strdel**(char \***s**)**

**{**

**char \***p**=**s**,\***q**;**

**int** count**=**0**;**

**while(\***p**!=**'\0'**) // 因为字符串以’\0’结束…**

**{**

**while(\***p**!=**' '**&&\***p**!=**'\0'**) //注意p!=’\0’不可去掉，不然最后一个单词结束不了了**

p**++; //若是字母就指针右移**

**if(\***p**==**' '**) //这句与前面的相对应，若是前面是空格，就执行**

**{ //如果是’\0’就不能执行if里面的了**

q**=**p**; //q接应p,完成后续移动**

count**++; //累计空格个数**

**while(\***q**!=**'\0'**) //移动过程，具体可参见插入排序**

**{**

**\***q**=\*(**q**+**1**);// //q指向元素的后面的全部左移**

q**++; //注意最后用’\0’完成终止**

**}**

**}**

**}**

**return** count**;**

**}**

**int main()**

**{**

**int** len**;**

**char** buf**[**81**];**

printf**(**"Please input a string:\n"**);**

gets**(**buf**);**

len**=**strdel**(**buf**);**

printf**(**"字符串中的空格个数为:%d\n"**,**len**);**

puts**(**"删除空格后字符串变为:"**);**

printf**(**"%s"**,**buf**);**

//printf("\n");

**return** 0**;**

**}**

4. 12年期末考题 10分

题目描述

编写函数void reverse(char\* s);

将形参s所指向的字符串反转（如s指向字符串为”hello world”，反转后为”dlrow olleh”）。编写main函数，在main中调用reverse函数，并输出反转后的字符串。(去年期末考题)

输入描述

输入字符串”hello world”，

输出描述

输出”dlrow olleh”

样例输入

hello world

样例输出

dlrow olleh

#include <stdio.h>

#include <string.h>

**void** reverse**(char \***s**,int** len**)// 基本思路是通过两个指针，头指针和尾指针**

**{ //让他们所指的值交换，直到~相遇~（len为奇数），**

**//或者~擦肩而过~(len为偶数)**

**char \***p1**=**s**,\***p2**=&**s**[**len**-**1**];**

**while(**p1**<**p2**)**

**{**

**\***p1**^=\***p2**; //这个是一种通过亦或位运算，实现交换的方法**

**\***p2**=\***p1**^\***p2**; // 原理是，亦或（^）这个位运算符 和他本身是互逆的**

**// \***p1**^=\***p2**; //** **自己可以试一下。其实这种方法有其局限性，**

**//如果\*p1=\*p2就挂掉了，因为已控制p1<p2,所以可以放心使用**

**//不想这样用的话，也可以这样**

**//char temp=\*p1;**

**// \*p1=\*p2;**

**// \*p2=temp;**

p1**++;**

p2**--;**

**}**

**}**

**int main()**

**{**

**char** str**[**81**],**ch**;**

**int** i**=**0**;**

**/\* while(**ch**=**getchar**(),**ch**!=**'\n'**)//或许看起来这样写很笨，**

**{ //是的，因为本来没想到gets就可以了**

**//后来知道了，也懒得改了**

str**[**i**++]=**ch**;**

**}**

str**[**i**]=**0**;\*/**

**gets(str); //上面的此一句可完美代替 不过也可以试着用别的方法做做**

reverse**(**str**,**i**);**

printf**(**"%s"**,**str**);**

**return** 0**;**

**}**

**附录：**

[关于 位运算](http://blog.csdn.net/superdullwolf/article/details/4649080)

[常去练习。。。](http://code.hdu.edu.cn/)