# C语言周测汇总

1. 阅读下面程序并写出输出结果（10分）。

main( )

{ int a[5]={1,2,3,4,5};

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

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

}

答：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. 用预处理指令#define声明一个常数，用以表明1年中有多少秒（忽略闰年问题）（5分）。

答：#define SECONDS\_PER\_YEAR (60 \* 60 \* 24 \* 365)UL

考点：   
1). #define 语法的基本知识（例如：不能以分号结束，括号的使用，等等）   
2). 懂得预处理器将为你计算常数表达式的值，因此，直接写出你是如何计算一年中有多少秒而不是计算出实际的值，是更清晰而没有代价的。   
3). 意识到这个表达式将使一个16位机的整型数溢出-因此要用到长整型符号L,告诉编译器这个常数是的长整型数。   
4).表达式中用到UL（表示无符号长整型）

1. 写一个"标准"宏MIN ，这个宏输入两个参数并返回较小的一个（5分）。

答：#define MIN(A,B)（（A）<= (B) ? (A): (B))

这个测试是为下面的目的而设的：   
1). 标识#define在宏中应用的基本知识。这是很重要的，因为直到嵌入(inline)操作符变为标准C的一部分，宏是方便产生嵌入代码的唯一方法，对于嵌入式系统来说，为了能达到要求的性能，嵌入代码经常是必须的方法。   
2). 三重条件操作符的知识。这个操作符存在C语言中的原因是它使得编译器能产生比if-then-else更优化的代码，了解这个用法是很重要的。   
3). 懂得在宏中小心地把参数用括号括起来

1. static 关键字的作用（10分）。

答：（1）static作用于局部变量，使得局部变量变成局部静态变量，改变了变量的生存期；（2）static作用于全局变量，使其变成静态全局变量，改变变量的作用域，使得其不可扩展声明到别的文件中，只能在本文件中调用；（3）static作用于外部函数，使其成为静态函数，使得函数不能扩展声明到别的文件中，只得 在本文件中调用。

1. 队列和栈有什么区别（10分）？

答：队列先进先出；栈先进后出。

1. 下面四个用const修饰指针有什么区别（10分）？

const int\* p;

int const\* p;

int\* const p;

const int\* const p;

答：前两个的作用是一样，a是一个常整型数。第三个意味着a是一个指向常整型数的指 针（也就是，整型数是不可修改的，但指针可以）。第四个意思a是一个指向整型数的常指针（也就是说，指针指向的整型数是可以修改的，但指针是不可修改 的）。最后一个意味着a是一个指向常整型数的常指针（也就是说，指针指向的整型数是不可修改的，同时指针也是不可修改的）。

1. 简述内存分区情况，说明各分区存放的什么样的数据（10分）。

答：代码区：存放函数二进制代码；

数据区：存放全局变量、静态变量、常量等；

堆区：程序员手动申请的；

栈区：局部变量、函数形参。

1. #include<filename>和#include”filename”有什么区别（10分）？

答：#include<filename>直接在库文件目录中搜索所包含的文件；#include”filename”先在当前目录下搜索所包含的文件。如果没有的话，再到库文件目录搜索。

1. 数组和链表的区别 （10分）。

答：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

1. 冒泡排序（10分）。

答：void BubbleSort(int arr[], int n)  
{  
     int i = 0, j =0;       
     for(i = 0; i < n; i++)  
       for(j = 0; j < n - 1 - i; j++)  
       {  
             if(arr[j] > arr[j + 1])  
             {  
                       arr[j] = arr[j] ^ arr[j+1];  
                       arr[j+1] = arr[j] ^ arr[j+1];  
                       arr[j] = arr[j] ^ arr[j+1];  
             }               
       }       
}

1. 写一个函数，交换两个整数，至少使用两种方法（10分）。

答：void Swap(int\* \_x,int\* \_y)

{

int nTmp = \*\_x;

\*\_x = \*\_y;

\*\_y = nTmp;

}

void Swap(int\* \_x,int\* \_y)

{

\*\_x = \*\_x ^ \*\_y;

\*\_y = \*\_x ^ \*\_y;

\*\_x = \*\_x ^ \*\_y;

}

T96每周一测（10.15）

C语言1—6章

1. 1、编写程序，实现以下功能。

用户输入12个数，存储到一个3行4列的数组中，然后找到其中的最大值和最小值，并且将其所在的行号和列号输出来

参考代码：

#include <stdio.h>

int main()

{

int array[3][4];

int i,j;

int max,min,maxI,maxJ,minI,minJ;

printf("input 12 numbers!\n");

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

{

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

scanf("%d",&array[i][j]);

}

max=array[0][0];

maxI=0;

maxJ=0;

min=array[0][0];

minI=0;

minJ=0;

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

{

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

{

if(array[i][j]>max)

{

max=array[i][j];

maxI=i;

maxJ=j;

}

if(array[i][j]<min)

{

min=array[i][j];

minI=i;

minJ=j;

}

}

}

printf("最大值：%d,i:%d,j:%d\n",max,maxI,maxJ);

printf("最小值：%d,i:%d,j:%d\n",min,minI,minJ);

return 0;

}

1. 编写程序，实现以下功能。

定义一个结构体，存储学生的信息，其中包括：学号，姓名，性别。开始时只有3个学生的信息，学号分别是1，2，3，姓名和性别自定。后来用户在第2个位置插入一个新生：学号4，姓名柳柳，性别女。然后显示插入后的全部学生信息。再后来，学号是2的学生转到其他班级了，所以要删掉，最后再显示删除后的全部学生信息。

参考代码：

#include <stdio.h>

struct Student

{

int no;

char name[20];

char sex;

}stu[10];

int length=3;

int main()

{

int i,j;

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

{

stu[i].no=i+1;

printf("input the no.%d student's name!\n",i+1);

scanf("%s",stu[i].name);

printf("input the no.%d student's sex!\n",i+1);

getchar();

scanf("%c",&stu[i].sex);

}

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

{

printf("the info of no.%d student!\n",i+1);

printf("学号：%d\n",stu[i].no);

printf("姓名：%s\n",stu[i].name);

printf("性别：%c\n",stu[i].sex);

}

//插入

for(i=2;i<length;i++)

{

stu[i+1]=stu[i];

}

stu[2].no=4;

stu[2].sex='n';

strcpy(stu[2].name,"liuliu");

length++;

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

{

printf("the info of no.%d student!\n",i+1);

printf("学号：%d\n",stu[i].no);

printf("姓名：%s\n",stu[i].name);

printf("性别：%c\n",stu[i].sex);

}

//删除

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

{

if(stu[i].no==2)

break;

}

if(i==length)

{

printf("没有学号是2的学生！\n");

exit(1);

}

else

{

for(j=i+1;j<length;j++)

{

stu[j-1]=stu[j];

}

length--;

}

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

{

printf("the info of no.%d student!\n",i+1);

printf("学号：%d\n",stu[i].no);

printf("姓名：%s\n",stu[i].name);

printf("性别：%c\n",stu[i].sex);

}

return 0;

}

1. 编写程序实现以下功能。

用户输入一个字符串，长度不超过100，写一个函数将其加密。加密方法为：当内容是英文字母时，用它在26字母中的后三个字母代替该字母，如：用d代替a,用a代替x，也就是循环代替。若为其它字符时不变。加密用1个函数，解密用一个函数（提醒，输入字符串、输出字符串都使用数组）

参考代码：

#include <stdio.h>

#include <string.h>

void encryption(char str[],int length)

{

int i;

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

{

if((str[i]>='a'&&str[i]<='w')||(str[i]>='A'&&str[i]<='W'))

str[i]+=3;

else if((str[i]>='x'&&str[i]<='z')||(str[i]>='X'&&str[i]<='Z'))

str[i]-=23;

}

printf("result:%s\n",str);

}

void decryption(char str[],int length)

{

int i;

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

{

if((str[i]>='d'&&str[i]<='z')||(str[i]>='D'&&str[i]<='Z'))

str[i]-=3;

else if((str[i]>='a'&&str[i]<='c')||(str[i]>='A'&&str[i]<='C'))

str[i]+=23;

}

printf("result:%s\n",str);

}

int main()

{

char str[101];

int i,length;

printf("input a string!\n");

scanf("%s",str);

length=strlen(str);

encryption(str,length);

decryption(str,length);

return 0;

}

1. 编写一个程序，从键盘上接收一个整数，并且从1-100之间找到能被它整除的整数，输出这些数以及这些数的个数。

参考代码：

#include <stdio.h>

int main()

{

int num,i,count=0;

printf("input a number!");

scanf("%d",&num);

for(i=1;i<=100;i++)

{

if(i%num==0)

{

printf("%d\n",i);

count++;

}

}

printf("个数：%d\n",count);

return 0;

}

16.从键盘上接收一个字符，判断它到底是哪种字符：数字，小写字母，大写字母，其他字符

#include "stdafx.h"

#include <stdio.h>

#include "stdafx.h"

#define PI 3.14

int main(void)

{

char ch\_t;

printf("请输入一个字符\n");

ch\_t = getchar();

if(ch\_t <='9' && ch\_t >='0')

printf("数字\n");

else if(ch\_t<= 'Z'&& ch\_t>='A')

printf("大写字母\n");

else if(ch\_t>='a'&& ch\_t<='z')

printf("小写字母\n");

else

printf("其他字符\n");

getchar();

getchar();

return 0;

}

1. 给出一百分制成绩，要求从键盘输入成绩后，输出成绩等级‘A’、‘B’、‘C’、‘D’、'E'。90分以上为‘A’,80—89分为‘B’，70—79分为‘C’，60—69分为‘D’，60分一下为'E'。

#include "stdafx.h"

#include <conio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

#include "stdafx.h"

int main(void)

{

int dSc;

printf("请输入需分级的分数。\n");

scanf("%d",&dSc);

if(dSc<=100)

{

if (dSc>=90)

printf("成绩%d分级为A",dSc);

else if(dSc>=80)

printf("成绩%d分级为B",dSc);

else if(dSc>=70)

printf("成绩%d分级为C",dSc);

else if(dSc>=60)

printf("成绩%d分级为D",dSc);

else if(dSc<60)

printf("成绩%d分级为E",dSc);

}

else

printf("输入的数据非法。\n");

getchar();

getchar();

return 0;

}

方法二

#include "stdafx.h"

int main(void)

{

int dSc;

int nStep;

printf("请输入需分级的分数。\n");

scanf("%d",&dSc);

nStep=dSc/10;

switch(nStep)

{

case 10: printf("成绩%d分级为A",dSc);

break;

case 9: printf("成绩%d分级为A",dSc);

break;

case 8: printf("成绩%d分级为B",dSc);

break;

case 7: printf("成绩%d分级为C",dSc);

break;

case 6: printf("成绩%d分级为D",dSc);

break;

default: printf("成绩%d分级为E。\n",dSc);

}

getchar();

getchar();

return 0;

}

1. 写出1一直加到100的程序，分别应用for，while和do while语句

#include "stdafx.h"

int main(void)

{

int na=1;

int sun=0;

while(na>=100){

sum+=sum+na;

na++;

}

printf(“%d”,sum);

return 0;

}

#include "stdafx.h"

int main(void)

{

int na=1;

int sun=0;

do{

sum+=sum+na;

na++;

}while(na>=100);

printf(“%d”,sum);

return 0;

}

#include "stdafx.h"

int main(void)

{

int na=1;

int sun=0;

for(na=1;na<=100;na++){

sum+=sum+na;

}

printf(“%d”,sum);

return 0;

}

1. （1）写出for，while和do while构成的死循环语句。

（2）写出while和do while的区别。

(1)

for(;;){}

while(1){}

do{}while(1);

(2)

do while是先执行再判断，至少执行一次。

While先判断，再执行。

1. （1）使用for循环输出1到50之间能被3整除的数

（2）使用for循环计算1到50之间的和，判断如果和大于344时推出程序，并打印出此时的和。

#include "stdafx.h"

int main(void)

{

int na=1;

int sun=0;

for(na=1;na<50;na++){

if(na%3==0){

printf(“%d”,na);

}

}

return 0;

}

(2)

#include "stdafx.h"

int main(void)

{

int na=1;

int sun=0;

for(na=1;na<50;na++){

sum+=sum+na;

if(sum>344){

printf(“%d”,na);

break;

}

}

return 0;

}

1. 设有定义：int n1=0 ,n2 , \*p=&n2, \*q=&n1;，以下赋值语句中与n2=n1;语句等价的是

**A)\*p=\*q;** B)p=q;

C)\*p=&n1; D)p=\*q;

1. 若有定义：int x=0, \*p=&x;，则语句printf("%d\n",\*p);的输出结果是

　　 A)随机值  **B)0**

　　 C)x的地址 D)p的地址

1. 以下定义语句中正确的是

　 　 A) char a='A'b='B'; B) float a=b=10.0;

**C) int a=10,\*b=&a;**  D) float \*a,b=&a;

1. 有以下程序

　　main()

　　{ int a=7,b=8,\*p,\*q,\*r;

　　p=&a;q=&b;

　　r=p; p=q;q=r;

　　printf("%d,%d,%d,%d\n",\*p,\*q,a,b);

　　}

　 　程序运行后的输出结果是

　　 A) 8,7,8,7 B) 7,8,7,8

**C) 8,7,7,8**  D) 7,8,8,7

1. 设有定义：int a,\*pa=&a;以下scanf语句中能正确为变量a读入数据的是

**A) scanf(“%d”,pa) ;**  B) scanf(“%d”,a) ;

C) scanf(“%d”,&pa) ; D) scanf(“%d”,\*pa) ;

1. 设有定义：int n=0,\*p=&n,\*\*q=&p;则以下选项中，正确的赋值语句是

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

1. 有以下程序片段：

　　void fun(char \*a, char \*b)

　　{ a=b; (\*a)++; }

　　main ()

　　{ char c1="A", c2="a", \*p1, \*p2;

　　p1=&c1; p2=&c2; fun(p1,p2);

　　printf(“&c&c\n”,c1,c2);

　　}

　　程序运行后的输出结果是

**A) Ab**  B) aa C) Aa D) Bb

1. 若程序中已包含头文件stdio.h, 以下选项中，正确运用指针变量的程序段是

A)int \*i=NULL; B) float \*f=NULL;

　　　 scanf("%d",i); \*f=10.5;

　　 C) char t=’m’, \*c=&t; 　　  **D) long \*L;**

　　　 \*c=&t; 　　**L=‘\0’;**

1. 有以下程序片段：

　　#inc1ude

　　main()

　　{ printf("%d\n", NULL); }

　　程序运行后的输出结果是

**A) 0**  B) 1 C) -1 D) NULL没定义，出错