**TEST – 09 函 数**

**测试 - 试题/答案**

【测试时间90分，自行批阅，计分并QQ告知 】

**学习和解题要点**

1.函数的概念和作用

在 Ｃ 程序设计中，为了能完成某种特定的功能，并且能适应不同的被加工数据，返回不同的结果，通常都用函数的形式来设计程序。一个较大的问题（程序），一般都被分解成若干个功能来完成，每个功能即一个函数－－程序模块，这也是结构化程序设计的思想。

在 Ｃ 语言中一般函数可写成：

〈数据类型〉 〈函数名〉( 〈形参１〉, 〈形参２〉, .....)

{ ........

........

return(〈返回值〉);

}

在某个地方要用函数时，可调用函数：〈函数名〉(〈实参１〉, 〈实参２〉, .....)

因此，函数可比仿成一个加工器（黑盒）：

加工 ( 函数 )

来料说明 ( 形参 )

产品 ( 函数返回值 )

比如函数是磨粉机，形参相当于来料说明，如可加工五谷，函数值是加工结果，如

磨成的粉。调用函数即开始使用磨粉机，实参用具体的大米，返回大米粉。实参用

小麦，返回面粉等等。

2.函数的定义

存储类型 （返回的）数据类型 函数名(形参说明语句序列) (**注：**此地分不能有分号)

{

函数内用的变量说明序列;(临时工作单元)

可执行语句序列;(包括 return(表达式)语句)

}

函数的返回值一般是通过 return 语句获得的。

返回的数据类型说明一般应该和return语句中的表达式类型一致，若不一致则以数据类型说明为准。如函数中没有 return 语句或 return；函数无返回值，返回一个用户很难确定的值。

说明：函数的数据类型：int char short long float double \*类型。

缺省为 int型。

不返回值为 void型。

函数的存储类型：内部函数 本文件内有效 static

外部函数 可被其他文件函数调用 extern

3．函数调用： 函数名(实参表)

可以为独立语句： 函数名(实参表); 表达式中： 变量＝函数名(实参表)＋其他式;

实参表与形参表的对应一般要做到三个一致：

个数一致

类型一致（在学指针时，指针变量和地址对应）

次序一致

实参都应该己预先说明的，并有确定的值。

一般，函数都应先定义后调用。对函数返回值是 int 或 char 的可先调用后定义。

如果非int 或 char型的函数一定要先调用后定义，则在函数调用语句前先作一个“申明”， 申明的方法是相当于把函数头部抄一遍，加一个分号：

存储类型 （返回的）数据类型 函数名(形参表)；(**注：**此地分号不可少)

4．函数调用的数据传递

把实参数据复制给形参，形参变量内数据变化不影响实参 －－值调用。

这是Ｃ语言函数最常规的调用方式

x 45

y 50

b 50

a 45

main()

{ int a=45,b=50;

c=max(a,b);

}

int max(int x,int y)

{ .............}

把实参的地址传给形参，这时形参变量应为指针变量，因此形参实参共用一个内存

位置，形参值的变化也即实参值的变化 －－地址调用（名调用）。

#include <stdio.h>

x

a

y

80

50

b

temp

50

main()

{ int a=50,b=80;

swap(&a,&b);

}

void swap(int \*x,int \*y)

{ int temp;

temp=\*x; \*x=\*y; \*y=temp;

}

这种方法可使函数除 return 返回值外，还可通过形参返回值。

数组与函数

指定的个别数组元素作实参时，是值调用。

在 Ｃ 语言中，由于数组名作为地址，因此实参和变参在以下四种情况时都为

地址调用： 实参和形参都用数组名

实参用数组名，形参用指针变量

实参和形参都用指针变量

实参用指针变量，形参为数组名

对后二种情况，实参指针都应先指向具体的数组。无论值调用还是地址调用，都是只能把实参的值传送给形参，形参的值不能传回给实参。

5. 变量的作用域和生存期

变量的作用域是指在程序中可使用（出现）该变量的范围。

有全局变量和局部变量。

变量的生存期是指（计算机）系统给变量分配内存空间到撒消空间的时段。

有分配以后固定存储空间（静态存储类别），直到程序结束才撒消。

有随时分配随时撤消的动态分配（动态存储类别）。

具体有四种 自动的（auto）静态的（static）寄存器的（register）

外部的（extern）

两者有联系又有区别，先要生存才能使用，但不能使用时并不是不存在。

总的来说下面几点是重要的：

函数的形参和函内定义的变量是局部变量，不在函数内定义的变量是全局变量。

只有在定义全局变量和局部静态变量时才能对数组初始化。

变量的生存期和作用域

内部变量

动态存储时段

生存期

定义点到程序结束

全局变量

定义该变量的函数

或复合语句内部

局部变量

作用域

静态存储时段

静态存储时段

外部变量

register

auto

省略(默认为auto)

省略(称无存储类型)

变量

允许的存储方式

static

static

6. 递归函数 函数体内，直接或间接地调用自已。最终有一个出口，不再调用自已。

用推展和回溯法读递归函数的计算结果：

int f( int n)

{ int y;

f(6)

2549 3\*f(5)+5

848 3\*f(4)+5

281 3\*f(3)+5

92 3\*f(2)+5

29 3\*f(1)+5

8 3\*f(0)+5

1

if (n>0)

y=3\*f(n-1)+5;

else y=1;

return(y);

}

当int a=6,z;z=f(a);

答案为 2549

**【 试 题 】**

一、单项选择题【10 X 1 = 10】

1. 在C语言程序中，有关函数的定义正确的是（ ）

A. 函数的定义可以嵌套，但函数的调用不可以嵌套

B. 函数的定义不可以嵌套，但函数的调用可以嵌套

C. 函数的定义和函数的调用均不可以嵌套

D. 函数的定义和函数的调用均可以嵌套

2．以下对C语言函数的有关描述中，正确的是（ ）。

A. 在C中，调用函数时，只能把实参的值传送给形参，形参的值不能传送给实参

B. C函数既可以嵌套定义又可递归调用

C. 函数必须有返回值，否则不能使用函数

D. C程序中有调用关系的所有函数必须放在同一个源程序文件中

3．函数调用语句“f((e1,e2),(e3,e4,e5));”中参数个数是（ ）。

A. 5 B. 4 C. 2 D. 1

4．C语言中函数的隐含存储类型是（ ）。

A. auto B. static C. extern D. 无存储类型

5．能把函数处理结果的二个数据返回给主调函数，在下面的方法中不正确的是（ ）

A. return 这二个数 B. 形参用二个元素的数组

C. 形参用二个这种数据类型的指针 D. 用二个全局变量

6. 有一函数的定义如：void fun(char \*s) { ........ },则不正确的函数调用是（ ）

1. main() B. main()

{ char a[20]=“abcdefgh”; { char a[20]=“abcdefgh”;

fun(a); fun(&a[0]);

...... ......

} }

C. main() D. main()

{ char a[20]=“abcdefgh”; { char a[20]=“abcdefgh”;

char \*p=a; fun(p); fun(a[]);

...... ......

} }

7．函数的功能是交换x和y中的值，且通过正确调用返回交换结果。不能正确执行此功能的函数是（ ）。

A. funa(int \*x, int \*y) B. funa(int x, int y)

{ int i,\*p=&i; { int t;

\*p=\*x; \*x=\*y; \*y=\*p; t=x; x=y; y=t;

} }

C. funa(int \*x, int \*y) D. funa(int \*x, int \*y)

{ int p; {

p=\*x; \*x=\*y; \*y=p; \*x=\*x+\*y; \*y=\*x-\*y; \*x=\*x-\*y;

} }

8．要求函数的功能是在一维数组a中查找x值；若找到则返回所在的下标值，否则返回0；数放在a[1]到a[n]中。不能正确执行此功能的函数是（ ）。

A. funa(int \*a, int n, int x) B. funb(int \*a, int n, int x)

{ \*a=x; { int k;

while(a[n]!=x) n--; for(k=1;k<=n;k++)

return(n); if(a[k]==x) return(k);

} return(0);

}

C. func(int a[],int n,int x) D. fund(int a[], int n, int x)

{ int \*k; { int k=0;

a[0]=x;k=a+n; do k++;

while(\*k!=x) k--; while((k<n+1)&&(a[k]!=x));

return(k-n); if((k<n+1)&&(a[k]==x)) return k;

} else return(0);

}

9．以下程序的输出结果是（ ）。

#include <stdio.h>

sub1(char a, char b) {char c; c=a; a=b; b=c;}

sub2(char \*a, char b) {char c; c=\*a; \*a=b; b=c;}

sub3(char \*a, char \*b) {char c; c=\*a; \*a=\*b; \*b=c;}

main()

{ char a,b;

a=’A’; b=’B’; sub3(&a,&b); putchar(a); putchar(b);

a=’A’; b=’B’; sub2(&a,b); putchar(a); putchar(b);

a=’A’; b=’B’; sub1(a,b); putchar(a); putchar(b);

}

A. BABBAB B. ABBBBA C. BABABA D. BAABBA

10.以下程序的输出结果是（ ）。

fun(char \*\*m)

{ ++m; printf(“%s\n”,\*m); }

main()

{ char \*a[]={ “BASIC”,”FOXPRO”,”C” };

fun(a);

}

A. BASIC B. ASIC C. FOXPRO D. C

二、填充题【10 X 2 = 20】

1．静态型内部变量的作用域是 。

2．函数中的形参和调用时的实参都是数组名时，传递方式为 ，都是变量时，传递方式为 。

3．函数的形式参数的作用域为 ，全局的外部变量和函数体内定义的局部变量重名时， 变量优先。

4．若自定义函数要求返回一个值，则应在该函数体中有一条 语句，若自定义函数要求不返回一个值，则应在该函数说明时加一个类型说明符 。

5. 若函数的形式参数是指针类型，则实参可以是 、 或 。

6．若给fun函数的形参s传递字符串：“∪∪6354abc”（其中“∪”表示空格字符），则函数的返回值是 。

#include <ctype.h>

long fun(char s[])

{ long n; int sign;

for(;isspace(\*s);s++);

sign=(\*s==’-‘)? -1:1;

if(\*s==’+’||\*s==’-‘) s++;

for(n=0;isdigit(\*s);s++)n=10\*n+(\*s-‘0’);

return(sign\*n);

}

7．下面函数要求计算两个整数x、y之和，并通过形参z传回该和值，请填空。

void add(int x, int y, )

{ =x+y; return; }

8．以下函数用来在w数组中插入x。n所指向的存储单元中存放w数组中字符个数。数组w中的字符已按从小到大的顺序排列，插入后数组w中的字符仍有序。请填空。

void fun(char \*w, char x, int \*n)

{ int i,p;

p=0;

w[\*n]=x;

while(x>w[p]) p++;

for(i=\*n;i>p;i--) w[i]= w[i-1] ;

w[p]=x;

++\*n;

}

9．函数my\_cmp()的功能是比较字符串s和t的大小，当s等于t时返回0，否则返回s和t的第一个不同字符的ASCII码差值，即当s>t时返回正值，当s<t时返回负值。请填空。

my\_cmp(char \*s, char \*t)

{ while(\*s==\*t)

{ if(\*s==’\0’) return(0);

++s; ++t;

}

return( );

}

10．以下程序是把字符串中的内容按逆序输出，但不改变串中的内容，请填空。

void inverp(char \*a)

{ if( ! ) return ;

inverp(a+1);

printf(“%c”, \* );

}

三、程序分析题【6 X 3 = 10】

1. 阅读函数，写出函数的主要功能。

ch(int \*p1, int \*p2)

{ int p;

if( \*p1>\*p2) { p=\*p1; \*p1=\*p2; \*p2=p; }

}

1. 阅读函数，写出函数的主要功能。

float av(float a[], int n)

{ int i; float s;

for(i=0,s=0;i<n;i++) s=s+a[i];

return(s/n);

}

1. 阅读下面程序，写出运算结果。

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));

}

1. 阅读下面程序，写出运算结果，并说明原因。

#include <stdio.h>

main()

{ int a[]={ 3,8,4,2 },i;

int x=0;

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

{ sub(a,x); printf(“%3d”,x); }

}

sub(int \*s, int y)

{ static int t=0;

y=s[t];

t++;

}

1. 阅读下面程序，写出运算结果，并说明函数的功能。

#include <stdio.h>

main()

{ int m,row,col;

int a[3][3]={ 50,-30,90,

35,45,-85,

-17,57,97};

func(a,&m,&row,&col,3,3);

printf(“%d,%d,%d\n”,m,row,col);

}

func(int a[][],int \*x,int \*y,int \*z,int m, int n)

{ int i,j,\*x=a[0][0];

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

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

if(a[i][j]<\*x)

{ \*x=a[i][j]; \*y=i; \*z=j;}

}

1. 阅读下面程序，写出运算结果。

#include <stdio.h>

main()

{ printf(“%5d\n”,fun(4)); }

fun (int n)

{ int t;

if((n==0)||(n==1)) t=3;

else t=n\*fun(n-1);

return t;

}

四、程序设计题 【13 X 4 = 52】

1. 编一个名为root的函数，求方程ax2+bx+c=0的b2-4ac，并作为函数的返回值。其中的a、b、c作为函数的形式参数。
2. 编一个函数，若有参数y为闰年，则返回1；否则返回0。
3. 编一个无返回值、名为root2的函数，要求如下：

形式参数：a,b,c单精度实型，root单精度实型数组名。

功能：计算ax2+bx+c=0的两个实根（设b2-4ac>0）存入root[2]中。

1. 编一个无返回值、名为max\_min的函数，对两个整数实参能求出它们的最大公约数和最小公倍数并显示。
2. 编一个能判断一个整数是否是素数的函数，并用它求出3到100之间的所有素数。
3. 编一个名为days的函数，要求如下：

形式参数：整数y,m,d分别表示年、月、日。

功能：计算该日是该年的第几天。

返回值：整数第几天。

1. 编一个无返回值、名为trus的函数，要求如下：

形式参数：s1[2][3],s2[3][2] 整型数组。

功能：将s1数组转置后存入s2数组中。

1. 编一个名为countc函数，要求如下：

形式参数：array存放字符串的字符型数组名。

功能：统计array数组中大写字母的数目。

返回值：字符串中大写字母的数目。

9. 编一个名为link函数，要求如下：

形式参数：s1[40],s2[40],s3[80]存放字符串的字符型数组。

功能：将s2连接到s1后存入s3中。

返回值：连接后字符串的长度。

10. 编一个函数，返回一维实型数组前n个元素的最大数、最小数和平均值。数组、n和最大数、最小数、平均值均作为函数的形式参数，本函数无返回值（用指针方法实现）。

11. 编一个函数，用“冒泡法”对字符数组中的字符按由小到大顺序排列。要求字符数组作为形参。

12. 编一函数，能将十六进制数转换成十进制数。

形参：字符指针，指向放十六进制数的字符数组

返回值：十进制整数

13．用递归法将一个整数转换成字符串。

【答案】

1. 单项选择题 【10 X 1 = 10】

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| B | A | C | C | A | D | B | C | A | C |

二、填充题 【10 X 2 = 20】

1. 所定义的函数或复合语句内

2. 地址传递 值传递

3. 函数内部 局部

4. return void

5. 变量地址 指针 数组名

6. fun函数中，先由for(;isspace(\*s);s++)；移过字符串中的先导空格字符，系统函数isspace(\*s)是判断空格字符的。若有字符’-‘,sign放-1，否则放1，移过字符’-‘或’+‘。从此取出字符串中的数字字符（系统函数isdigit(\*s)是判断数字字符的）其他字符不要，变成整数后，按位乘10相加(n=n\*10+(\*s-‘0’))，最后返回数sign\*n值。 6354

7. 由于不用return返回值，因此z必须为指针 int \*z ，调用时用add(a,b,&c)使z指向实参c，相加结果放 \*z =x+y 即放实参c。

8. w中从0算起有n指向数个字符，则w[\*n]是空的，先把x中的字符放w[\*n]。找出按序x中字符应放的位置p。(while(x>w[p]) p++)，从最后一个字符w[i=\*n](即要插入的字符)开始向前移位( w[i]= w[i-1] )直到位p，则就把最后一个字符(即要插入的字符)插入到了p位。

9. \*s-\*t

10. 该递归函数的出口是if，因此，在此要填入遇到字符串结束标志时返回。字符串结束标志的方式有四种：\*a==’\0’,\*a==NULL,\*a==0 或 !\*a 。由于if中已有求反运算符!，故可填入： \*a 或填(\*a!=’\0’),(\*a!=NULL),(\*a!=0)。 显然%c \* 后要填入 a .

三、程序分析题 【6 X 3 = 18】

1. 函数的功能是能把p1和p2所指的实参数据交换。

2． 函数的功能：求出n个实数的平均值

3． 运算结果：把一个无符号整数一位一位取出相乘。结果为12

4． 程序运算结果: 0 0 0 0 , 这是因为sub(a,x)的调用x对应y，是值调用,y的值并不返回给x，四次调用x都是0。

5． 程序运算结果: -85, 1, 2

该函数的功能是求二维数组中最小元素值及它的行列下标。由于有三个数据要返回，因此用三个整数指针，而调用这个函数时，用了三个整数的地址。

6． 求fun(4)，可用推展和回溯法读递归函数计算结果为：72

四、程序设计题【13 X 4 = 52】

1. 解

float root(float a, float b, float c)

{ return(b\*b-4\*a\*c); }

2．解

int year(int y)

{ if((y%4==0&&y%100!=0)||y%400==0) return 1;

else return 0;

}

3．解

#include <math.h>

void root2(float root[2], float a, float b, float c)

{ float p;

p=sqtr(b\*b-4\*a\*c);

root[0]=(-b+p)/(2\*a);

root[1]=(-b-p)/(2\*a);

}

1. 解

void max\_min(int m, int n)

{ int a,b,t,r;

a=m; b=n;

if (m<n) { t=m; m=n; n=t;}

r=m%n;

while(r!=0)

{ m=n; n=r;

r=m%n;

}

printf(“%d 和 %d 的最大公约数是 %\n”,a,b,n);

printf(“%d 和 %d 的最小公倍数是 %\n”,a,b,a\*b/n);

}

5. 解

#include <stdio.h>

#include <math.h>

int prime(int m)

{ int k,i;

k=sqrt(m);

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

if(m%i==0) return 0;

return 1;

}

void main()

{ int a;

for(a=3;a<=100;a++)

if (prime(a)) printf(“%3d”,a);

printf(“\n”);

}

6. 解

int days(int y, int m, int d)

{ int n;

switch(1)

{ case 1: n=0; if(m==1) break;

case 2: n=n+31; if(m==2) break;

case 3: n=n+28; if(m==3) break;

case 4: n=n+31; if(m==4) break;

case 5: n=n+30; if(m==5) break;

case 6: n=n+31; if(m==6) break;

case 7: n=n+30; if(m==7) break;

case 8: n=n+31; if(m==8) break;

case 9: n=n+31; if(m==9) break;

case 10: n=n+30; if(m==10) break;

case 11: n=n+31; if(m==11) break;

case 12: n=n+30;

}

n=n+d;

if(m>2)

if((y%4==0&&y%100!=0)||(y%400==0)) n=n+1;

return n;

}

1. 解

void trus(int s1[2][3], int s2[3][2])

{ int i,j;

for(i=0;i<2;i++)

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

s2[j][i]=s1[i][j];

}

1. 解

include <string.h>

int countc(char array[])

{ int i, n=0;

for(i=0;i<strlen(array);i++)

if(array[i]<=’A’&& array[i]>=’Z’) n++;

return n;

}

1. 解

include <string.h>

int link(char s1[40], char s2[40], char s3[80])

{ int i, k, n=0;

for(i=0;i<strlen(s1);i++)

{ s3[i]=s1[i]; n++; }

k=i;

for(i=0;i<strlen(s2);i++)

{ s3[k+i]=s2[i]; n++; }

s3[i]=’\0’;

return n;

}

1. 解

void fun(float a[], int n, float \*max, float \*min, float \*vag)

{ int i;

\*vag=a[0]; \*max=a[0]; \*min=a[0];

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

{ if(a[i]>\*max) \*max=a[i];

if(a[i]<\*min) \*min=a[i];

\*vag=\*vag+a[i];

}

\*vag=\*vag/n;

}

11．解

void sortc(char a[])

{ int n, i, j;

char ch;

n=strlen(a);

for(i=0;i<n-1;i++)

for(j=0;j<n-i-1;j++)

if(a[j]>a[j+1])

{ ch=a[j]; a[j]=a[j+1]; a[j+1]=ch;}

}

12．解

#include <ctype.h>

int tv(char \*s)

{ int m,n=0;

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

{ if(isalpha(\*s)) \*s=toupper(\*s);

switch(\*s)

{ case 'F': m=15; break;

case 'E': m=14; break;

case 'D': m=13; break;

case 'C': m=12; break;

case 'B': m=11; break;

case 'A': m=10; break;

default: m=\*s-48;

}

n=n\*16+m; s++;

}

return n;

}

13．解

#include <string.h>

char a[255]; //全局数组

void contw(int m) //递归转换

{ int n;

static i=1;

char c;

if(m!=0){

n=m%10; c=n+48; a[i]=c;

m=m/10; i++;

contw(m);

}

return;

}

void convert(char \*b) //倒置数组

{ int l,i;

char c, \*p;

l=strlen(b); p=b+l-1;

for(i=1,b++;i<=l/2;i++,b++,p--)

{ c=\*b; \*b=\*p; \*p=c;}

return ;

}

void main()

{ int n;

void contw( int m);

void convert(char \*b);

printf("输入一整数:");

scanf("%d",&n);

if(n<0) {a[0]='-'; n=-n;}

else a[0]=' ';

contw(n);

convert(a);

printf("字符串: %s\n",a);

return;

}