**2022年程序设计基础8**

**一．单选题**

1. 在一个被调用函数中，关于 return 语句使用的描述，错误的是（ D）。

A.被调用函数中可以不用 return 语句

B.被调用函数中可以使用多个 return 语句

C.被调用函数中，如果有返回值，就一定要有 return 语句

D.被调用函数中，一个 return 语句可以返回多个值给调用函数

2．将两个字符串连接起来组成一个字符串时，选用函数（C ）。

A.strlen( ) B.strcpy( ) C.strcat( ) D.strcmp( )

3．数组名作为参数时，以下叙述正确的是（C ）。

A.函数调用时是将实参数组的所有元素值传给形参。

B.函数调用时是将实参数组的第一个元素值传给形参。

C.函数调用时是将实参数组的首地址传给形参。

D.函数调用时是将实参数组的所有元素的地址传给形参。

4．以下程序的运行结果是（C ）。

int x = 5, y = 6;

void incxy( )

{

x++;

y++;

}

int main(void )

{

int x = 3;

incxy( );

printf("%d, %d\n", x, y);

return 0;

}

A.3, 6 B.4, 7 C.3, 7 D.6, 7

5.函数调用语句function((a,b),c)中含有的实参个数为（C ）。

A.1 B.3 C.2 D.语法有误

6.当一个函数无返回值时，函数的返回类型应为( B)。

A.任意 B.void C.int D.char

**二、程序填空题**

1.输出 150 到 200之间有且仅有一位数字为9的所有整数。要求定义和调用函数is(n, digit)判断正整数n是否有且仅有一位数字为digit，若满足条件则返回1，否则返回0。  
运行示例：  
159 169 179 189 190 191 192 193 194 195 196 197 198

#include <stdio.h>

int main(void)

{

int i;

int is(int n,int digit);

for(i = 150; i <= 200; i++)

if( is(i,9) )

printf("%d ", i);

printf("\n");

return 0;

}

int is(int n,int digit)

{

int number,count=0;

do{

number = n % 10;

if( number==digit ) count++; 2分

n = n / 10; 2分

}while(n != 0);

if(count==1) return 1; 2分

else return 0;

}

2．变量地址做函数参数

老师想写一个用于交换变量x,y的值的swap 函数。

#include <stdio.h>

void swap(int a, int b);

int main() {

int x;

int y;

x = 20;

y = 16;

printf("x = %d\ny = %d\n", x, y);

swap(x, y);

printf("x = %d\ny = %d\n", x, y);

return 0;

}

void swap(int a, int b) {

int temp;

temp = a;

a = b;

b = temp;

}

执行结果为：



为什么呢？由于在main函数中调用swap时，是将实参x的值复制给了a，将实参y的值复制给了b，swap函数中的形参a和实参x存储在不同的内存地址中，形参b和实参y也存储在不同的内存地址中。

因此swap函数内的a、b的交换并不影响到main函数中x,y的值。所以老师的这个程序并不能按照预期工作。

不知道小伙伴们还记得不？我们学习scanf时，在变量名前要加取地址符&，很多小伙伴忘记，在PTA提交答案的时候系统给出了段错误。为什么呢?

通过取地址符我们可以知道一个变量所对应在内存中的位置。

如果在这里，我们能不直接传入x的值和y的值，而是让函数传它们在内存中的位置，那么我们就可以在函数中对这两个地址进行访问和操作了。

那我们一起来修改下这个代码吧。

第一步：修改main函数中调用swap函数时传递的实际参数。

请将下面代码中调用swap函数时传入的参数x和y改为它们对应的内存地址。（第二个空）

第二步：修改函数声明和函数实现。

传入到swap函数的实际参数从整型变量变成整数型变量的地址，相应的，我们也需要对函数的形参的类型进行变更。

我们可以通过int \*p声明一个变量，用来存储整型变量x（已经定义：int x;）的地址值：

int \*p;

p=&x;

类似的，为了接收整型变量x和y的地址，swap函数中的形式参数也相应的应该是可以用来存储整型变量地址值的变量。

请将swap函数的形式参数的类型改变一下，让两个形式参数可以接收整数型变量的地址。（在函数的声明和实现部分都要修改）（第一、三空）

第三步：完成swap函数，实现交换实参x，y。

这里我们需要用到取值运算符：它和乘号的写法一致，都是\*，但是它用于存储地址的变量前。

比如，对于一个存储整数型变量的的地址的变量p，我们如果希望取出它存储的地址对应的内存的值赋值给另一个变量m，我们可以直接写m=\*p;

而如果我们希望将p存储的地址对应的内存的值改为3，我们则可以直接写\*p=3;

如果我们不改变变量temp的类型，我们可以在哪些地方增加取值运算符∗用以完成交换传入参数值的任务呢？动手做一做吧。

#include <stdio.h>

void swap( int \*a, int \*b ); 2分

int main() {

int x;

int y;

x = 20;

y = 16;

printf("x = %d\ny = %d\n", x, y);

swap( &x, &y ); 2分

printf("x = %d\ny = %d\n", x, y);

return 0;

}

void swap( int \*a, int \*b ) {

int temp;

temp = \*a ; 2分

\*a =\*b ; 2分

\*b = temp; 2分

}

3．编写一个函数，使给定的一个二维数组（3X3）转置，即行列互换。

#include<stdio.h>

#define N 3

void main()

{

void convert(int array[3][3]) ; 2分

int arr[N][N]={{1,2,3},{4,5,6},{7,8,9}};

int i,j;

convert(arr);

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

{for(j=0;j<N;j++)

printf("%3d",arr[i][j]);

printf("\n");

}

}

void convert(int array[3][3])

{

int i,j,t;

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

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

{

t=array[i][j];

array[i][j]=array[j][i];

array[j][i]=t ; 2分

}

}

4．使用函数分段计算居民水费[2]

为鼓励居民节约用水，自来水公司采取按用水量阶梯式计价的办法，居民应交水费y（元）与月用水量x（吨）相关：当x不超过15吨时，y=4x/3；超过后，y=2.5x−10.5。请编写程序实现水费的计算，要求定义和调用函数f(x)计算水费，其中x的类型是double，函数类型是double。

#include <stdio.h>

double f(double x); 2分

int main()

{

double x, y;

scanf ("%lf", &x);

y = f(x); 2分

printf ("f(%.2f) = %.2f\n", x, y);

return 0;

}

double f(double x)

{

double result;

if (x <= 15){

result = 4 \* x / 3;

}else{

result = 2.5 \* x - 10.5;

}

return result; 2分

}

5．计算二维数组中奇数元素之和

下面的程序是计算二维数组中奇数元素之和的程序。请填写空缺的代码。

#include <stdio.h>

int sumOdd(int (\*array)[5], int row)

{

int sum= 0 ;

for(int i=0; i<row; i++)

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

if( array[i][j]%2==1|array[i][j]%2 ) sum += array[i][j] ;

return sum;

}

int main(){

int array[4][5];

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

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

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

printf("%d\n", sumOdd( array,4 )); //注意：填空处两端均已有括号，只需填函数参数

return 0;

}

6．统计字符串中最长的单词

输入一个字符串，统计其中最长的单词并输出。

#include <stdio.h>

int findLongest( char str[] );

int main()

{

char sArr[100] = { 0 };

int loc, i;

gets(sArr);

loc = findLongest(sArr) ; 2分

for (i = loc; sArr[i] != ' ' && sArr[i] != '\0'; i++)

putchar(sArr[i]);

return 0;

}

int findLongest( char str[] )

{

int i = 0, Loc = 0, len = 0, lLoc = 0;

while (str[i] != '\0')

{

while (str[i] == ' ')

i++;

lLoc = i;

while (str[i] != ' ' && str[i] != '\0') i++ ; 2分

if (i -lLoc > len)

{

len = i – lLoc ; 2分

Loc = lLoc;

}

}

return Loc ; 2分

}

###输入样例

This is a C Program

###输出样例

Program

**三、编程题**

**1.  圆形体体积计算器**

本题要求实现一个常用圆形体体积的计算器。计算公式如下：

* 球体体积 *V*=34​*πr*3，其中*r*是球体半径。
* 圆柱体体积 *V*=*πr*2*h*，其中*r*是底圆半径，*h*是高。
* 圆锥体体积 *V*=31​*πr*2*h*，其中*r*是底圆半径，*h*是高。

### 输入格式：

在每次计算之前，要求输出如下界面：

|  |
| --- |
| 1-Ball  2-Cylinder  3-Cone  other-Exit  Please enter your command: |

然后从标准输入读进一个整数指令。

### 输出格式：

如果读入的指令是1或2或3，则执行相应的体积计算；如果是其他整数，则程序结束运行。

* 当输入为1时，在计算球体体积之前，打印Please enter the radius:，然后读入球体半径，完成计算；
* 当输入为2时，在计算圆柱体体积之前，打印Please enter the radius and the height:，然后读入底圆半径和高，完成计算；
* 当输入为3时，在计算圆锥体体积之前，打印Please enter the radius and the height:，然后读入底圆半径和高，完成计算。

计算结果在一行内输出，保留小数点后两位。

### 输入样例：

|  |
| --- |
| 1  2  3  2.4 3  0 |

### 输出样例：

|  |
| --- |
| 1-Ball  2-Cylinder  3-Cone  other-Exit  Please enter your command:  Please enter the radius:  33.51  1-Ball  2-Cylinder  3-Cone  other-Exit  Please enter your command:  Please enter the radius and the height:  18.10  1-Ball  2-Cylinder  3-Cone  other-Exit  Please enter your command: |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **#include<math.h>**  **#define PI 3.1415926535**  **int main()**  **{**  **int a;**  **double h, r;**  **do {**  **printf("1-Ball\n2-Cylinder\n3-Cone\nother-Exit\nPlease enter your command:\n");**  **scanf("%d", &a);**  **if (a == 1) {**  **printf("Please enter the radius:\n");**  **scanf("%lf", &r);**  **printf("%.2lf\n", 4 \* PI \* pow(r, 3) / 3);**  **}**  **else if (a == 2) {**  **printf("Please enter the radius and the height:\n");**  **scanf("%lf%lf", &r, &h);**  **printf("%.2lf\n", PI \* r \* r \* h);**  **}**  **else if (a == 3) {**  **printf("Please enter the radius and the height:\n");**  **scanf("%lf%lf", &r, &h);**  **printf("%.2lf\n", PI \* r \* r \* h / 3);**  **}**  **} while (a == 1 || a == 2 || a == 3);**  **return 0;**  **}** |

**2．整数分解为若干项之和**

将一个正整数N分解成几个正整数相加，可以有多种分解方法，例如7=6+1，7=5+2，7=5+1+1，…。编程求出正整数N的所有整数分解式子。

### 输入格式：

每个输入包含一个测试用例，即正整数N (0<N≤30)。

### 输出格式：

按递增顺序输出N的所有整数分解式子。递增顺序是指：对于两个分解序列*N*1​={*n*1​,*n*2​,⋯}和*N*2​={*m*1​,*m*2​,⋯}，若存在*i*使得*n*1​=*m*1​,⋯,*ni*​=*mi*​，但是*ni*+1​<*mi*+1​,则*N*1​序列必定在*N*2​序列之前输出。每个式子由小到大相加，式子间用分号隔开，且每输出4个式子后换行。

### 输入样例：

|  |
| --- |
| 7 |

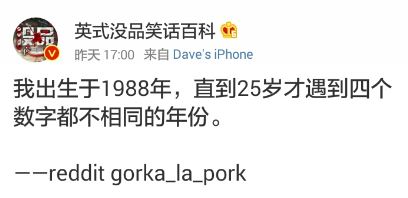
### 输出样例：

|  |
| --- |
| 7=1+1+1+1+1+1+1;7=1+1+1+1+1+2;7=1+1+1+1+3;7=1+1+1+2+2  7=1+1+1+4;7=1+1+2+3;7=1+1+5;7=1+2+2+2  7=1+2+4;7=1+3+3;7=1+6;7=2+2+3  7=2+5;7=3+4;7=7 |

### 参考代码：

|  |
| --- |
| #include<stdio.h>int cnt=0;int a[50];void f(int number,int n,int length,int sum){if(sum==n){cnt++;printf("%d=",n);printf("%d",a[0]);for(int i=1;i<length;i++) printf("+%d",a[i]);if(cnt%4==0&&cnt!=0) printf("\n");if(cnt%4!=0&&number!=n) printf(";");}if(sum>n) return;for(int i=number;i<=n;i++){a[length]=i;f(i,n,length+1,sum+i);}}int main(){int n;scanf("%d",&n);f(1,n,0,0);} |

3．**出生年**



以上是新浪微博中一奇葩贴：“我出生于1988年，直到25岁才遇到4个数字都不相同的年份。”也就是说，直到2013年才达到“4个数字都不相同”的要求。本题请你根据要求，自动填充“我出生于y年，直到x岁才遇到n个数字都不相同的年份”这句话。

### 输入格式：

输入在一行中给出出生年份y和目标年份中不同数字的个数n，其中y在[1, 3000]之间，n可以是2、或3、或4。注意不足4位的年份要在前面补零，例如公元1年被认为是0001年，有2个不同的数字0和1。

### 输出格式：

根据输入，输出x和能达到要求的年份。数字间以1个空格分隔，行首尾不得有多余空格。年份要按4位输出。注意：所谓“n个数字都不相同”是指不同的数字正好是n个。如“2013”被视为满足“4位数字都不同”的条件，但不被视为满足2位或3位数字不同的条件。

### 输入样例1：

|  |
| --- |
| 1988 4 |

### 输出样例1：

|  |
| --- |
| 25 2013 |

### 输入样例2：

|  |
| --- |
| 1 2 |

### 输出样例2：

|  |
| --- |
| 0 0001 |

### 参考代码：

|  |
| --- |
| #include <stdio.h>#include <math.h>#include <string.h>int main(){int a[4];int year, n;scanf("%d%d",&year, &n);int i, count = 0;for(i=year; i<3020; i++){count = 1;a[0] = i/1000;a[1] = i/100%10;a[2] = i/10%10;a[3] = i%10;if(a[0] != a[1] && a[0] != a[2] && a[0] != a[3]) count++;if(a[1] != a[2] && a[1] != a[3]) count++;if(a[2] != a[3]) count++;if(count == n) break;} |

4．**判断数字字符**

编写函数，判断指定的字符是否是数字字符，如果是函数返回1，否则函数返回0.要求在主调函数中输入该字符，调用函数并输出判断结果。

**输入格式:**

在一行中给出1个字符。

**输出格式:**

对于给定输入的字符，输出判断结果yes或no。

**输入样例:**

|  |
| --- |
| 6 |

**输出样例:**

|  |
| --- |
| yes |

**输入样例:**

|  |
| --- |
| H |

**输出样例:**

|  |
| --- |
| no |

**参考代码：**

|  |
| --- |
| **#include <stdio.h>**  **int judge(char ch);**  **int main(){**  **char ch;**  **scanf("%c",&ch);**  **int ret = judge(ch);**  **if(ret==1) printf("yes");**  **else printf("no");**  **return 0;**  **}**  **int judge(char ch){**  **if(ch>='0'&&ch<='9')**  **return 1;**  **else**  **return 0;**  **}** |

**5．任意进制下的可逆素数**

任意进制下的可逆素数是这样定义的：它自身是一个素数，当它转换为任意进制之后，把所有数字逆序，得到的新数字的值仍然是一个素数。比如，73在十进制下是一个素数，它在十进制下的逆37也是一个素数，那就称它是十进制下是一个可逆素数。

### 输入格式:

输入有多行，每行包括两个正整数，数N以及基R，其中N<100000, 1<D<=10, 遇到负数则退出。

### 输出格式:

对每一行输入，如果是可逆素数，则输出“Yes”，否则输出“No”

### 输入样例:

在这里给出一组输入。例如：

|  |
| --- |
| 73 10  23 2  23 10  -2 |

### 输出样例:

在这里给出相应的输出。例如：

|  |
| --- |
| Yes  Yes  No |

样例说明：23的二进制是10111，其逆为11101，对应的十进制数是29，所以是可逆素数。

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **#include<math.h>**  **int Prime(int n);**  **void Reverse(int n1, int n2);**  **int main() {**  **int n1, n2;**  **while (scanf("%d",&n1), n1 > -1) {**  **scanf("%d", &n2);**  **if (Prime(n1)) Reverse(n1, n2);**  **else printf("No\n");**  **}**  **return 0;**  **}**  **int Prime(int n) {**  **int d=1;**  **if(n==1)d=0;**  **for (int i = 2; i <= sqrt(n); i++)**  **if (n % i == 0) { d = 0; break;}**  **return d;**  **}**  **void Reverse(int n1, int n2) {**  **int d,t=0,s1=0,s2=0;**  **while (n1) {**  **s1 += n1 % n2 \* pow(10, t++);**  **n1 /= n2;**  **}**  **while (s1) {**  **s2 += s1 % 10 \* pow(n2, --t);**  **s1 /= 10;**  **}**  **d = Prime(s2);**  **if (d == 1)printf("Yes\n");**  **else printf("No\n");**  **}** |

**6．静态局部变量分析代码**

静态局部变量程序举例,分析代码

#include<stdio.h>

long next(long n){

//static long s=0;

long s=0;

if(n%2==1) n=n\*3+1;

else n=n/2;

s++;

printf("Times of %ld is %ld.\n",s,n);

return n;

}

int main(){

long n;

scanf("%ld",&n);

while(n!=1){

n=next(n);

}

return 0;

}

### 输入样例:

|  |
| --- |
| 5 |

### 输出样例:

|  |
| --- |
| Times of 1 is 16.  Times of 1 is 8.  Times of 1 is 4.  Times of 1 is 2.  Times of 1 is 1. |

**参考代码：**

|  |
| --- |
| **#include<stdio.h>**  **long next(long n){**  **long s=0;**  **if(n%2==1) n=n\*3+1;**  **else n=n/2;**  **s++;**  **printf("Times of %ld is %ld.\n",s,n);**  **return n;**  **}**  **int main(){**  **long n;**  **scanf("%ld",&n);**  **while(n!=1){**  **n=next(n);**  **}**  **return 0;**  **}** |