9.4逆序输出

输入一个不多于5位的正整数，要求： （1）求出它是几位数 分别输出每一位数字 按逆序输出各位数字，例如原数为321，应输出123

10.1

输入两个正整数m和n，求其最大公约数和最小公倍数

#include "stdio.h"

int main()

{

int p,r,n,m,temp;

scanf("%d %d",&n,&m);

if (n <m) {

temp=n;

n=m;

m=temp;

}

p=n\*m;

while(m!=0) {

r=n%m;

n=m;

m=r;

}

printf("最大公约数:%d\n",n);

printf("最小公倍数:%d",p/n);

return 0;

}

10.2

输出所有的“水仙花数”，所谓“水仙花数”是指一个3位数，其各位数字立方和等于该数字本身。

#include "stdio.h"

int main()

{

int i,j,k,n;

for (n=100; n <1000; n++) {

i=n/100;

j=n/10-i\*10;

k=n%10;

if (n==i\*i\*i + j\*j\*j + k\*k\*k)

printf("%d ",n);

}

printf("\n");

return 0;

}

10.3

一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如，6的因子为1，2，3，且6=1+2+3，因此6是“完数”。编程找出1000之内的所有完数，按如下格式输出其因子：6 its factors are 1 2 3

#include "stdio.h"

int main() {

int m,s,i;

for (m=2; m <1000; m++) {

s=0;

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

if ((m%i)==0) s=s+i;

if(s==m) {

printf("%d,its factors are ",m);

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

if (m%i==0) printf("%d ",i);

printf("\n");

}

}

return 0;

}

10.4

找出所有满足以下特性的6位整数：它是一个完全平方数（完全平方数是指该数是某个整数的平方），该数同时也是回文数（回文数是指该数等于它的反序数）。例如，698896是836的平方并且698896是回文数，因此698896是满足所给条件的6位整数。输出所有满足上述条件的6位整数。、

#include "stdio.h"

#include "math.h"

int main()

{

int i,j,k;

for(i=100000;i <=999999;i++){

j=i;

k=0;

while(j!=0){

k=k\*10+j%10;

j=j/10;

}

if(i==k&&(int)sqrt(i)==sqrt(i))

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

}

}

10.5

求1!+2!+…+20!，其中x!=1\*2\*…\*x，表示阶乘。

#include <stdio.h>

int main()

{

int i,n;

unsigned long long sum=1;

unsigned long long count=0;

scanf("%d",&n);

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

sum=sum\*i;

count+=sum;

}

printf("%ld",count);

return 0;

}

11.1

输出一个以下的杨辉三角形，要求输出10行 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

#include "stdio.h"

#define N 10

int main()

{

int i,j,a[N][N];

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

a[i][i]=1;

a[i][0]=1;

}

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

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

a[i][j]=a[i-1][j-1]+a[i-1][j];

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

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

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

printf("\n");

}

return 0;

}

11.2

求一个3\*3的整型矩阵主对角线元素之和 1,2,3 4,5,6 7,8,9

#include "stdio.h"

int main()

{

int a[3][3]={{1,2,3},{4,5,6},{7,8,9}},sum=0;

int i,j;

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

sum=sum+a[i][i];

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

return 0;

}

11.3

用选择排序法对10个整数进行排序[1,12,4,9,10,22,-7,0,99,8]

#include "stdio.h"

int main()

{

int i,j,min,temp,a[11]={1000,1,12,4,9,10,22,-7,0,99,8};

for (i=1; i <=9; i++) {

min=i;

for (j=i+1; j <=10; j++)

if (a[min]>a[j]) min=j;

temp=a[i];

a[i]=a[min];

a[min]=temp;

}

for (i=1; i <=10; i++)

printf("%d ",a[i]);

return 0;

}

11.4

有一个已排好序的数组{1,4,6,9,13,16,19,28,40,100}，要求输入一个数后，按原来排序的规律将它插入数组中。

#include "stdio.h"

int main()

{

int a[11]= {1,4,6,9,13,16,19,28,40,100};

int temp1,temp2,number,end,i,j;

scanf("%d",&number);

end=a[9];

if (number>end)

a[10]=number;

else {

for (i=0; i <10; i++) {

if (a[i]>number) {

temp1=a[i];

a[i]=number;

for (j=i+1; j <11; j++) {

temp2=a[j];

a[j]=temp1;

temp1=temp2;

}

break;

}

}

}

for (i=0; i <11; i++)

printf("%d ",a[i]);

return 0;

}

11.5

键盘输入一个长度为10的int型数组，删除数组中重复出现的数据 如，原数组是： 1 4 2 3 4 1 2 5 5 9，输出：1 4 2 3 5 9

#include "stdio.h"

int main(void)

{

int a[10],i,j;

for(i=0;i <10;i++){

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

}

for(i=0;i <10;i++){

if(a[i]!=-1){

for(j=i+1;j <10;j++)

if(a[i]==a[j]) a[j]=-1;

}

}

for(i=0;i <10;i++)

if(a[i]!=-1) printf("%d ",a[i]);

return 0;

}

12.1

有一篇文章，共有3行文字，每行小于80个字符。要求分别统计出其中英文大写字母、小写字母、数字、空格以及其他字符的个数

#include "stdio.h"

int main()

{

int i,j,upp,low,dig,spa,oth;

char text[3][80];

upp=low=dig=spa=oth=0;

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

gets(text[i]);

for (j=0; j <80 && text[i][j]!='\0'; j++) {

if (text[i][j]>='A'&& text[i][j] <='Z')

upp++;

else if (text[i][j]>='a' && text[i][j] <='z')

low++;

else if (text[i][j]>='0' && text[i][j] <='9')

dig++;

else if (text[i][j]==' ')

spa++;

else

oth++;

}

}

printf("upper case:%d\n",upp);

printf("lower case:%d\n",low);

printf("digit:%d\n",dig);

printf("space:%d\n",spa);

printf("other:%d",oth);

return 0;

}

12.2

输入一行电文，已按下面规律译成密码： A->Z a->z B->Y b->y C->X c->x 即第1个字母变成第26个字母，第i个字母变成第(26-i+1)个字母，非字母字符不变。要求编程序将密码译回原文，并输出原文。

#include "stdio.h"

int main()

{

int j,n;

char ch[80],tran[80];

gets(ch);

j=0;

while (ch[j]!='\0') {

if ((ch[j]>='A') && (ch[j] <='Z'))

tran[j]=155-ch[j];

else if ((ch[j]>='a') && (ch[j] <='z'))

tran[j]=219-ch[j];

else

tran[j]=ch[j];

j++;

}

n=j;

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

putchar(tran[j]);

printf("\n");

return 0;

}

12.3

编一程序，将两个字符串连接起来，不要用strcat函数

#include <stdio.h>

int main()

{

char s1[80],s2[40];

int i=0,j=0;

scanf("%s",s1);

scanf("%s",s2);

while (s1[i]!='\0')

i++;

while(s2[j]!='\0')

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

s1[i]='\0';

printf("%s",s1);

return 0;

}

12.4

编一个程序，将两个字符串s1和s2比较，若s1>s2，输出一个正数；若s1=s2，输出0；若s1<s2，输出一个负数。不要用strcpy函数。两个字符串用gets函数读入。输出的正数或负数的绝对值是相比较的两个字符串第一个不同的相应字符的ASCII码的差值。例如“A”与“C”比，由于“A”<“C”，应输出负数，同时由于两者ASCII码差值为2，因此应输出-2。同理，“And”和“Aid”比较，根据第二个字符比较结果，‘n’比‘i’大5，因此输出5

#include "stdio.h"

int main()

{

int i,resu;

char s1[100],s2[100];

scanf("%s",s1);

scanf("%s",s2);

i=0;

while ((s1[i]==s2[i]) && (s1[i]!='\0')) i++;

if (s1[i]=='\0' && s2[i]=='\0')

resu=0;

else

resu=s1[i]-s2[i];

printf("result:%d",resu);

return 0;

}

13.1

写2个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用这两个函数，并输出结果。两个整数由键盘输入。

#include "stdio.h"

int main()

{

int hcf(int,int);

int lcd(int,int,int);

int u,v,h,l;

scanf("%d %d",&u,&v);

h=hcf(u,v);

printf("hcf=%d\n",h);

l=lcd(u,v,h);

printf("lcd=%d",l);

return 0;

}

int hcf(int u,int v) {

int t,r;

if (v>u) {

t=u;

u=v;

v=t;

}

while ((r=u%v)!=0) {

u=v;

v=r;

}

return(v);

}

int lcd(int u,int v,int h) {

return(u\*v/h);

}

13.2

给出年、月、日，计算该日是该年的第几天

#include "stdio.h"

int main()

{

int sum\_day(int month,int day);

int leap(int year);

int year,month,day,days;

scanf("%d %d %d",&year,&month,&day);

days=sum\_day(month,day);

if(leap(year)&&month>=3)

days=days+1;

printf("%d",days);

return 0;

}

int sum\_day(int month,int day) {

int day\_tab[13]= {0,31,28,31,30,31,30,31,31,30,31,30,31};

int i;

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

day+=day\_tab[i];

return(day);

}

int leap(int year) {

int leap;

leap=year%4==0&&year%100!=0||year%400==0;

return(leap);

}

13.3

寻找[0,100]区间内所有的孪生素数并输出。孪生素数是指差为2的两个素数，例如，3和5，5和7。

编写函数int prime(int n)，判断素数。

#include <stdio.h>

int main()

{

int test(int i);

int i,j;

for(i=1;i<=100;i++)//如果求[m,n]区间的孪生素数则改为for(i=m;i<=n;i++)

{

if(test(i)==1)

{

j=i+2;

if(test(j)==1)

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

}

}

}

int test(int i){

int j;

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

{

if(i%j==0)

break;

}

if(j<i) return 0;

else return 1;

}

13.4

哥德巴赫猜想：任何一个大于4的偶数都可以表示为两个素数之和。验证[6,50]之间的偶数

写函数void guest(int n)，找到并输出所有素数对，每个素数对之和均等于偶数n。

编写main函数，遍历[6,50]间所有的偶数，调用guest函数查找并输出素数对。

输出格式如下：

40=3+37 40=11+29 40=17+23（换行）

#include "stdio.h"

#include "math.h"

int prime(int n)

{

int i;

for(i=2;i <=sqrt((double)n);i++){

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

}

return 1;

}

void guest(int n)

{

int i;

for(i=3;i <=n/2;i+=2){

if(prime(i)&&prime(n-i))

printf("%d=%d+%d ",n,i,n-i);

}

putchar('\n');

}

int main(void)

{

int m=6,n=50,i,t=0;

for(i=m;i <=n;i+=2){

guest(i);

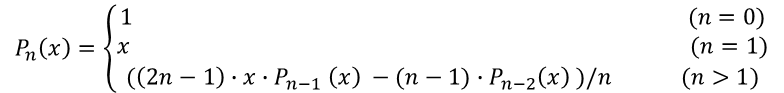
}

return 0;

}

14.1

采用递归方法求下面多项式： 其中n和x为任意正整数。在主函数中输入数据并调用函数得到结果。



#include <stdio.h>

double p(int n,int x){ //注意双精度

if (n==0){ //if语句判断n属于哪一个代数式

return 1;

} else if (n==1) {

return x;

} else {

return ((2\*n-1)\*x\*p((n-1),x)-(n-1)\*p((n-2),x))/n/1.0; //返回多项式值

}

}

int main()

{ int n, x;

scanf("%d%d",&n,&x);

printf("%.2f",p(n,x)); //保留两位小数

return 0;

}

14.2

写一个函数，使给定的一个3\*3的二维整型数组转置，即行列互换。

在main函数中：给定下列数组，调用自定义函数，并在main函数中输出转换后的二维数组。

原数组：

1 2 3

3 4 5

4 5 6

#include <stdio.h>

#define N 3

int main()

{

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

int i,j;

int array[N][N]={{1,2,3},{3,4,5},{4,5,6}};

convert(array);

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

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

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

printf("\n");

}

return 0;

}

void convert(int array[][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;

}

}

14.3

写一个函数，使输入的一个字符串按反序存放，在主函数中输入和输出字符串

#include "stdio.h"

#include "string.h"

int main()

{

char str[100];

scanf("%s",str);

inverse(str);

printf("%s",str);

return 0;

}

void inverse(char str[]) {

char t;

int i,j;

for (i=0,j=strlen(str); i <(strlen(str)/2); i++,j--) {

t=str[i];

str[i]=str[j-1];

str[j-1]=t;

}

}

14.4

写一个函数，其功能是：将一个字符串中的元音字母复制到另一个字符串。

在主函数中输入字符串，调用自定义函数，然后在主函数中输出处理后的字符串。

#include "stdio.h"

int main()

{

void cpy(char [],char []);

char str[80],c[80];

gets(str);

cpy(str,c);

printf("%s",c);

return 0;

}

void cpy(char s[],char c[]) {

int i,j;

for (i=0,j=0; s[i]!='\0'; i++)

if (s[i]=='a'||s[i]=='A'||s[i]=='e'||s[i]=='E'||s[i]=='i'||s[i]=='I'||s[i]=='o'||s[i]=='O'||s[i]=='u'||s[i]=='U') {

c[j]=s[i];

j++;

}

c[j]='\0';

}

14.5

写一个函数，对于一行字符（含空格），能够找到字符串中最长的单词。

主函数中，输入字符串，调用自定义函数并输出最长的单词。

比如输入：a aa aaa aaaa

输出：aaaa

#include <stdio.h>

#include <string.h>

int main()

{

int alphabetic(char);

int longest(char []);

int i;

char line[100]={0};

gets(line);

for (i=longest(line); alphabetic(line[i]); i++)

printf("%c",line[i]);

printf("\n");

return 0;

}

int alphabetic(char c) //判断是否为字母符号

{

if ((c>='a' && c <='z')||(c>='A'&&c <='z'))

return(1);

else

return(0);

}

int longest(char string[]) //寻找最长单词的第一个字母在字符串中的序号

{

int len=1,i,length=0,flag=1,place=0,point;

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

if (alphabetic(string[i]))

if (flag) {

point=i;

flag=0;

} else

len++;

else {

flag=1;

if (len>=length) {

length=len;

place=point;

}

len=1;

//printf("%d %d\n",place,length); //for debug only

}

return(place);

}

15.1

编写函数计算一维实型数组前n个元素的最大值、最小值和平均值。数组、n、最大值、最小值和平均值均作为函数形参，函数无返回值；在主函数中输入数据，调用函数得到结果。（要求用指针方法实现）

#include "stdio.h"

void f(double A[],int n,double \*max,double \*min,double \*aveg){

\*aveg=\*max=\*min=A[0];

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

\*aveg=\*aveg+A[i];

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

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

}

\*aveg=\*aveg/n;

}

int main()

{ int n;

scanf("%d",&n);

double A[n],max,min,aveg;

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

scanf("%lf",&A[i]);

f(A,n,&max,&min,&aveg);

printf("max:%.2lf\nmin:%.2lf\naveg:%.2lf",max,min,aveg);

return 0;

}

15.2

有5个整数，使前面各数顺序向后移m个位置，最后m个数变成最前面m个数，写一个函数实现以上功能，在主函数中输入5个整数和输出调整后的5个数

#include "stdio.h"

int main()

{

void move(int [20],int,int);

int number[20],n=5,m,i;

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

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

scanf("%d",&m);

move(number,n,m);

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

printf("%d ",number[i]);

printf("");

return 0;

}

void move(int array[20],int n,int m) {

int \*p,array\_end;

array\_end=\*(array+n-1);

for (p=array+n-1; p>array; p--)

\*p=\*(p-1);

\*array=array\_end;

m--;

if (m>0) move(array,n,m);

}

15.3

在主函数中，从键盘上输⼊5个数据到⼀维数组中。编写函数int max(int \*p)，功能为：找出数组中的最⼤值所在的元素下标。在主函数中调用max函数，并输出数组中的最⼤值所在的元素下标。

#include<stdio.h>

//需要引入的文件

//如提示运行通过，但解题失败，认证对比预期输出和实际输出之间的差异

int max(int \*p)

{

int max;

int i;

int index;

max = \*p;

index = 0;

for(i=1;i<5;i++)

{

if(max < \*(p+i))

{

max = \*(p+i);

index = i;

}

}

return index;

}

int main()

{

int a[5];

int i;

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

{

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

}

printf("最大值下标:%d\n",max(a));

return 0;

}

15.4

编写函数fun(int \*p1,int \*p2,int \*p3)，其功能是：求p1所指一维数组中的所有能被5整除的数的个数以及这些所有能被5整除的数的积；分别将所求得的个数以及积通过指针p2和p3返回主函数。

主函数中，由键盘输入3个整数存入一维数组, 调用函数fun后, 在主函数输出最后的结果。

例如:

输入: 10 2 5

输出: 2  50

# include <stdio.h>

void fun(int \*p1,int \*p2,int \*p3)

{int i;

for(i=0;i<3;i++,p1++)

{

if((\*p1)%5==0)

{

(\*p2)++;

\*p3=(\*p3)\*(\*p1);

}

}

}

int main()

{

int a[3];

int i, p2=0,p3=1;

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

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

fun(a,&p2,&p3);

printf("%d %d",p2,p3);

return 0;

}

小测试2

编写函数insert(char str1[], char str2[],int n)，其功能是：将str2数组里的字符串插入字符数组str1内第n个字符之后。

在main函数中，分别输入两个字符串（str1和str2，最大长度都小于10)、一个整型数字n(不大于str1的实际长度)，调用函数insert，在main函数中输出插入合并后的字符串。【需至少两条注释】

【提示】：算法关键点在于移动str1中第n个之后的字符，需考虑避免字符的覆盖

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

int main()

{

void insert(char str1[], char str2[],int n);

char str1[100]={0},str2[100]={0}; //定义字符串，并初始化为空字符

int n;

scanf("%s",str1);

scanf("%s",str2);

scanf("%d",&n);

insert(str1,str2,n); //调用insert函数，实现字符串插入

puts(str1);

return 0;

}

void insert(char str1[], char str2[],int n)

{

int i,len1,len2;

len1=strlen(str1);

len2=strlen(str2);

for (i=1; i<len1-n+1; i++) //循环次数len1-n，即把str1中从序号len1-1到len1-n+1逐个移动len2个位置

str1[len1+len2-i]=str1[len1-i];

for (i=0; i<len2; i++) //循环次数len2,即把str2中各元素插入str1中

str1[n+i]=str2[i];

}

16.1

使用字符指针编写程序，输入一个长度为n的字符串a，在字符串a的i （0＜i＜n）处插入字符x，输出插入后的字符串a。（n，x，i的值可自由输入)。

#include "stdio.h"

#include "string.h"

int main()

{ int i;

char a[100]={0},x,\*p=a;

gets(p);

scanf("%d",&i);

x=getchar();

for (p=a+strlen(a)-1;p>=a+i;p--)

\*(p+1)=\*p;

\*(++p)=x;

puts(a);

return 0;

}

16.2

编写函数intsearch(char \*cpSource, char ch)，该函数在一个字符串中找到可能的最长的子字符串，该字符串是由同一字符组成的。从主函数中输入"aabbcccddddeeeeeffffff"和'e'，调用函数得到结果。

#include "stdio.h"

#include "string.h"

int search(char \*cpSource,char ch){

int max=0,cnt=0;

for (int i=0;i <=strlen(cpSource);i++){

if (cpSource[i]==ch)

cnt++;

else{

if (cnt>max) max=cnt;

cnt=0;

}

}

return max;

}

int main()

{ char S[100],ch;

gets(S);

ch=getchar();

int max=search(S,ch);

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

putchar(ch);

return 0;

}

16.3

输入字符串，要求将每个单词的第一个字母改为大写字母，并输出。 如“hello world！”改为“Hello World”

#include "stdio.h"

void main() {

char s[255],prrior=' ',\*p;

gets(s);

for(p=s; \*p!='\0'; p++) {

if(prrior==' '&& \*p>='a'&& \*p <='z')

\*p-=32;

prrior=\*p;

}

puts(s);

}

16.4

编写函数replace(char \*str,char \*fstr,char \*rstr)，将str所指字符串中凡是与fstr字符串相同的字符替换成rstr（rstr与fstr的字符长度不一定相同）。从主函数中输入原始字符串"iffordowhileelsewhilebreak"、查找字符串"while"和替换字符串"struct"，调用函数得到结果。

#include "stdio.h"

#include "string.h"

void replace(char \*str,char \*fstr,char \*rstr){

char dest[100]={0};

int flag=0,i,j,k;

for (i=0; i <strlen(str);i++){

for (j=i,k=0;fstr[k]==str[j];k++,j++);

if (fstr[k]=='\0') {

flag=1 ;

}

if (flag==1){//找到一个源子串，用目的子串替换

strcat(dest,rstr);

i=i+k-1;

flag=0;

}else{//没找到源子串，将源串中该部分字符复制到目的串

int l=strlen(dest);

dest[l]=str[i];

}

}

puts(dest);

strcpy(str,dest);

}

int main()

{ char str[100],fstr[10],rstr[10];

gets(str);

gets(fstr);

gets(rstr);

replace(str,fstr,rstr);

puts(str);

return 0;

}

17.1

有3个候选人，每个选民只能投票选一人，要求编一个统计选票的程序，先后输入被选人的名字，最后输出各人得票结果。

#include <string.h>

#include <stdio.h>

struct Person //声明结构体类型struct Person

{ char name[20]; //候选人姓名

int count; //候选人得票数

} leader[3]={"Li",0,"Zhang",0,"Sun",0}; //定义结构体数组并初始化

int main()

{ int i,j;

char leader\_name[20]; //定义字符数组

for(i=1;i<=10;i++)

{ scanf("%s",leader\_name); //输入所选的候选人姓名

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

if(strcmp(leader\_name,leader[j].name)==0) leader[j].count++;

}

printf("\nResult:\n");

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

printf("%5s:%d\n",leader[i].name,leader[i].count);

return 0;

}

小测试3

在main函数中，通过键盘分别读入字符串（最大长度不超3 0包含字母、数字)和一个字符，利用自定义函数 findlet对字符串进行处理，然后在main函数中输出 ：该字符在字符数组中第一次出现的位置（下标值）（若没有该字符，则输出-1）。

函数 void findlet(char \*str, char \* let, int \*order) 的功能：针对指针str指向的字符数组，找到指针let所指向的字符在该字符数组中第一次出现的位置（下标值）（若没有该字符，则输出-1），并把该下标值存储到指针order所指向的变量。如字符串 “12a89aaRry6ru”中，字母a第一次出现的下标值为2。而字母b没有出现，下标值为-1。

【要有至少两条注释】

示例：

输入：

12a89aaRry6ru

a

输出：

2

#include <stdio.h>

#include <string.h>

int main()

{

int findlet(char \*str,char \*let,int \*order);

char a[50],ch;

int order; //存储下标

gets(a);

scanf("%c",&ch);

findlet(a,&ch,&order);

printf("%d",order);

return 0;

}

int findlet(char \*str,char \*let,int \*order)

{

int n,k,j;

j=-1; //下标初值设为-1

n= strlen(str);

for(k=0;k<n;k++,str++)

if(\*str==\*let) //判断是否相等

{j=k;

break;}

\*order=j;

}