1.LB4202 求任意的一个m×m矩阵(2<=m<=20)的最大数及其所在的行列数

#include <stdio.h>

int main(void)

{

int i, j, m, max, maxi, maxj;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

max = array[0][0];

maxi=0;maxj=0;

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

{

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

{

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

{

max = array[i][j];

maxi = i;

maxj = j;

}

}

}

printf("Output:");

printf("\nmax=%d,i=%d,j=%d\n", max, maxi, maxj);

return 0;

}

2.LB4203 从键盘上读入一行字符(约定：字符数≤127字节)，按以下方法将其加密变换：

#include <stdio.h>

int main(void)

{

int i;

char string[128];

printf("Please input string:");

gets(string);

for (i = 0; string[i] != '\0'; i++)

{

if (string[i] >= 'a' && string[i] <= 'z')

{

string[i] = 'a' + 25 - (string[i] - 'a');

}

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

{

string[i] = 'A' + 25 - (string[i] - 'A');

}

}

printf("Output:\nzi fu chuan chang du:%d", i);

printf("\nmi wen:%s\n", string);

return 0;

}

3.HH4205 比较字符串

输入两个字符串，比较两字符串的长度大小关系。

#include <cstdio>

#include <cstring>

int main() {

int t,k;

char s1[100], s2[100];

scanf("%d", &t);

while (t--) {

scanf("%s %s", s1, s2);

printf("Output:\n");

k = strlen(s1) - strlen(s2);

if (k > 0)

printf("%s is longer than %s\n", s1, s2);

else

printf("%s %s %s\n", s1, 0 == k ? "is equal long to" : "is shorter than", s2);

}

return 0;

}

4.HH4211 字符串插入空格

写一个函数，输入一个4位数字，要求输出这4个数字字符，但每两个数字间空一个空格。

#include <stdio.h>

int main() {

void output(char digits[]);

char str[5];

printf("Input:\n");

gets(str);

printf("Output:\n");

output(str);

return 0;

}

void output(char digits[]) {

int i;

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

if (i > 0)

printf(" ");

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

}

puts("");

}

5.WB4209 查找并删除指定字符串

给定一个短字符串（不含空格），再给定若干字符串，在这些字符串中删除所含有的短字符串。<

#include <stdio.h>

#include <string.h>

char \* Strlwr(char \*str){

int i=0;

while(str[i]){

if('A'<=str[i] && str[i]<='Z'){

str[i] += 32;

}

i++;

}

return str;

}

int main(){

char subStr[1000]; // 用来存储需要删除的字符串

char str[1000]; // 用来存储待删除的整串

char strTmp[1000];

int i=0, j=0;

int subLen;

printf("请输入要被删除的字符串：\n");

gets(subStr); // 读取需要删除的字符串

Strlwr(subStr);

subLen = strlen(subStr); // 记录需要删除的字符串的长度

printf("请输入待删除的字符串：\n");

gets(str); // 获取整串

strcpy(strTmp, str);

Strlwr(strTmp);

printf("Output:\n被删除指定字符串后的字符串如下：\n");

while(str[i]){

if(str[i] == ' '){

i++; // 如果当前是空格，则跳过这个空格

}else if(strncmp(strTmp+i, subStr, subLen) == 0){

i += subLen; // 如果当前连续 subLen 个字符时需要删除的字符串，则跳过这个长度

}else{

str[j++] = str[i++]; // 如果不是上面两种情况，则拷贝字符

}

}

str[j] = '\0'; // 最后字符需要结尾

puts(str); // 输出结果

return 0;

}

6.WB4210 计算个单词有多少个字母

编一个程序，读入用户输入的，以“.”结尾的一行文字，统计并分别输出每个单词含有多少个字符。

#include <stdio.h>

#include <string.h>

int main(){

char str[1000]; // 用来读取整个字符串

char \*pStr;

int iCount; // 一个字符指针，用来存储以空格和点号分割的字符串

printf("Please input some sentences,end of each sentence with '.':\n");

gets(str);

pStr = strtok(str, " ."); // 先用空格和点号作一次切割

iCount = 0; // 记录已经有多少数字输出了，由于每个数字间用一个空格分开，因而需要记录数字的输出数目

printf("Output:\nEach words' letter number is :\n");

while(pStr){

if(iCount++){

putchar(' '); // 数字间使用一个空格分割

}

printf("%d", strlen(pStr)); // 输出单词的长度

pStr = strtok(NULL, " ."); // 做下一次分割

}

putchar('\n'); // 别忘了换行

return 0;

}

7.LB4208 s1输入为"123456789"，S2输入为"abcdefghijk"，则输出的s1为：1a2b3c4d5e6f7g8h9ijk

#include <stdio.h>

#include <string.h>

int main(void)

{

int i, k;

char s1[128], s2[128], s3[256] = {'\0'};

printf("Please input string1:");

gets(s1);

printf("Please input string2:");

gets(s2);

for (i = 0, k = 0; s1[i] != '\0'&&s2[i] != '\0'; i++)

{

s3[k] = s1[i];

k++;

s3[k] = s2[i];

k++;

}

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

{

strcat(s3, &s2[i]);

}

else

{

strcat(s3, &s1[i]);

}

printf("Output:\nstring1:%s\n", s3);

return 0;

}

8.WB4208 查找输入目标字符数

<p>统计一个给定字符串中指定的字符出现的次数。先输入要查找的字符，可包含空格和特殊字符。

#include <stdio.h>

int main(){int i = 0;

char str1[10], str2[100];

printf("请输入要统计的字符（可包含空格）：\n");

gets(str1);

printf("请输入被查找的字符串：\n");

gets(str2);

printf("Output:\n查找结果：\n");

while(str1[i]){

int j=0, count=0;

while(str2[j]){

if(str1[i] == str2[j]){

count++;

}

j++;

}

printf("%c %d\n", str1[i], count);

i++;

}

return 0;

}

9.LB4302 实现十进制转num换成R进制数

#include <stdio.h>

#include <string.h>

void fun(char str[40], int num, int rad);

int main(void)

{

int num, rad;

char str[40];

printf("Input the num, R: ");

scanf("%d,%d", &num, &rad);

fun(str, num, rad);

printf("Output:\n");

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

return 0;

}

void fun(char str[40], int num, int rad)

{

int i = 0, j;

int flag = 0;

char str2[40];

char number[17] = "0123456789ABCDEF";

if (num == 0)

{

strcpy(str, "0");

return ;

}

if (num < 0)

{

flag = 1;

num = -num;

}

while (num > 0)

{

str2[i] = number[num % rad];

i++;

num = num / rad;

}

str2[i] = '\0';

j = strlen(str2) - 1;

if (flag == 0)

{

i = 0;

}

else

{

str[0] = '-';

i = 1;

}

for (; j >= 0; i++, j--)

{

str[i] = str2[j];

}

str[i] = '\0';

}

10.HH4302 火车票购票

火车经过X站，火车最大载客人数为m，有n个订票请求，请求订购从a站到b站的k张票，若能满足订购要求则输出1，否则输出0。

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int cmp(const void \*a,const void \*b) {

return \*(int \*)a-\*(int \*)b;

}

int unique(int a[],int n) {

int i,j;

if (!n) return 0;

qsort(a,n,sizeof(a[0]),cmp);

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

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

}

return j;

}

int bf(int a[],int n,int val) {

int l=0,r=n;

while(l<r) {

if (a[(l+r)/2]<val) l=(l+r)/2+1; else r=(l+r)/2;

}

return l;

}

int d[1<<19][2];

int query(int t,int l,int r,int a,int b) {

if (l==a && r==b) return d[t][1];

if (d[t][0]) {

d[t<<1][0]+=d[t][0];

d[t<<1][1]+=d[t][0];

d[t<<1|1][0]+=d[t][0];

d[t<<1|1][1]+=d[t][0];

d[t][0]=0;

}

if (b<=(l+r)/2) return query(t<<1,l,(l+r)/2,a,b);

else if (a>=(l+r)/2) return query(t<<1|1,(l+r)/2,r,a,b);

else {

int r1=query(t<<1,l,(l+r)/2,a,(l+r)/2);

int r2=query(t<<1|1,(l+r)/2,r,(l+r)/2,b);

if (r2>r1) return r2; else return r1;

}

}

void update(int t,int l,int r,int a,int b,int val) {

if (l==a && r==b) {

d[t][0]+=val;

d[t][1]+=val;

return ;

}

if (d[t][0]) {

d[t<<1][0]+=d[t][0];

d[t<<1][1]+=d[t][0];

d[t<<1|1][0]+=d[t][0];

d[t<<1|1][1]+=d[t][0];

d[t][0]=0;

}

if (b<=(l+r)/2) update(t<<1,l,(l+r)/2,a,b,val);

else if (a>=(l+r)/2) update(t<<1|1,(l+r)/2,r,a,b,val);

else {

update(t<<1,l,(l+r)/2,a,(l+r)/2,val);

update(t<<1|1,(l+r)/2,r,(l+r)/2,b,val);

}

if (d[t<<1][1]>d[t<<1|1][1]) d[t][1]=d[t<<1][1];

else d[t][1]=d[t<<1|1][1];

}

int x[200000];

int a[100000][3];

int n,m;

int main() {

int i,j,k;

printf("Input:\n");

while(scanf("%d %d",&n,&m)!=EOF) {

memset(d,0,sizeof(d));

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

scanf("%d %d %d",&a[i][0],&a[i][1],&a[i][2]);

a[i][1]++;

x[i\*2]=a[i][0],x[i\*2+1]=a[i][1];

}

printf("Output:\n");

k=unique(x,n+n);

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

a[i][0]=bf(x,k,a[i][0]);

a[i][1]=bf(x,k,a[i][1]);

j=query(1,0,k,a[i][0],a[i][1]);

if (j+a[i][2]>m) puts("0");

else {

puts("1");

update(1,0,k,a[i][0],a[i][1],a[i][2]);

}

}

exit(0);

}

return 0;

}

;11.LB4201 排序格式输出，以-222结束

#include <stdio.h>

int main(void)

{

int i, j, cnt, temp;

int array[10000];

printf("Please input numbers:");

for (i = 0; i < 10000; i++)

{

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

if (array[i] == -222)

{

break;

}

}

cnt = i;

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

{

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

{

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

{

temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

printf("\nOutput:\n");

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

{

if (i == cnt - 1)

{

printf("%-6d\n", array[i]);

break;

}

if ((i + 1) % 6 != 0)

{

printf("%-6d,", array[i]);

}

else if ((i + 1) % 6 == 0)

{

printf("%-6d", array[i]);

printf("\n");

}

}

return 0;

}

12.WB4207 对子字符串进行排序

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

char str[1024];

int idx[1024];

int cmp(const void \*a,const void \*b) {

return strcmp(str+\*(int \*)a,str+\*(int \*)b);

}

int main() {

int i;

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

gets(str);

for(i=0;str[i];i++) idx[i]=i;

qsort(idx,i,sizeof(int),cmp);

printf("Output:\nThe sorted strings are:\n");

for(i=0;str[i];i++) printf("%s\n",str+idx[i]);

return 0;

}

13.D4204 ;猪安家

<p>通过猪的安家计算有多少头猪：Andy和Mary养了很多猪。他们想要给猪安家。

#include <stdio.h>

int main(void)

{

int i, j;

int n, sum, sum1;

int chengji, chengjisum;

int result;

int chushu[10], yushu[10];

printf("please input how many time he changed the piggery:\n");

scanf("%d", &n);

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

{

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

}

sum = 0;

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

{

chengji = 1;

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

{

chengji = chengji \* chushu[j];

}

chengjisum = chengji;

chengji = chengji / chushu[i];

sum1 = chengji;

while (sum1 % chushu[i] != yushu[i])

{

sum1 = sum1 + chengji;

}

sum = sum + sum1;

}

while (sum >= 0)

{

sum = sum - chengjisum;

}

result = sum + chengjisum;

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

return 0;

}

14.HH4212 &nbsp;&nbsp;&nbsp;当总统

<h3>题目描述</h3><p>小明想当丑国的总统，丑国大选是按各州的投票结果来确定最终的结果的

#include<stdio.h>

int main()

{

int n,i,j,t,s,a[101];

printf("input:\n");

scanf("%d",&n);

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

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

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

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

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

{

t=a[j];

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

a[j+1]=t;

}

for(s=i=0;i<n/2+1;i++)

s+=a[i]/2+1;

printf("Output:\n%d\n",s);

}

15.LB4104 用键盘输入的整数产生5\*5矩阵N，并按行输出该矩阵，每个元素占4个数位、右对齐

#include <stdio.h>

int main(void)

{

int n, i, j;

int array[5][5];

printf("Please input an integer:");

scanf("%d", &n);

printf("Output:\n");

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

{

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

{

array[i][j] = n + i + j;

printf("%4d", array[i][j]);

}

printf("\n");

}

return 0;

}

16.HE4206 求上网

对于每组输入，如果小明所在的位置能连上网，则输出“YES”，否则输出“NO”

#include <stdio.h>

int main() {

int n, i;

double x[100], y[100], r[100], a, b;

printf("Please input n:");

scanf("%d", &n);

printf("Output:\n");

while (n) {

printf("Input date:\n",n);

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

scanf("%lf%lf%lf", &x[i], &y[i], &r[i]);

scanf("%lf%lf", &a, &b);

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

if ((a - x[i]) \* (a - x[i]) + (b - y[i]) \* (b - y[i]) <= r[i]

\* r[i])

break;

}

puts(i < n ? "YES" : "NO");

printf("Please input n:");

scanf("%d", &n);

}

return 0;

}

17.HH4202 Problem B

<p>请写一个程序，对于一个m行m列的（1&lt;m&lt;10）的方阵，求其每一行，每一列及主对角线元素之和，最后按照从大到小的顺序依次输出。

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int cmp(const void \*a,const void \*b) {

return \*(int \*)b-\*(int \*)a;

}

int main() {

int a[32];

int n,i,j,t;

while(scanf("%d",&n)!=EOF) {

memset(a,0,sizeof(a));

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

scanf("%d",&t);

a[i]+=t;

a[j+n]+=t;

if (i==j) a[n+n]+=t;

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

}

qsort(a,n+n+2,sizeof(a[0]),cmp);

printf("Output:");

for(i=0;i<n+n+1;i++) printf("%d ",a[i]);

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

exit(0);

}

return 0;

}

18.LB4204 输入10个整型数存入一维数组，输出值和下标都为奇数的元素及个数

#include <stdio.h>

int main(void)

{

int i, count = 0;

int array[10], barray[10];

printf("Input 10 integers.\n");

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

{

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

}

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

{

if (array[i] % 2 == 1 && i % 2 == 1)

{

barray[count] = i;

count++;

}

}

printf("Output:\ncount=%d\n", count);

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

{

printf("a[%d]=%d\n", barray[i], array[barray[i] ]);

}

return 0;

}

;19.HH4303 复数集合

id="des\_7f3f029d875d48b2bee69773e19c6cd5">

<p>一个复数（x+iy）集合，两种操作作用在该集合上：

#include<stdio.h>

#include<string.h>

int main()

{

int complex[1000][3],x,y;

char str[100];

int mmax=0,amax=0,bmax=0;

int p=0,n=0,i=0,j=0,k=0,size=0;

scanf("%d",&n);

size=0;

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

{

scanf("%s",str);

printf("Output:\n");

if(strcmp(str,"Pop")==0)

{

if(size==0)

{

printf("empty\n");

}

else

{

mmax=0,amax=0,bmax=0,p=0;

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

{

if((mmax<complex[j][2])||((mmax==complex[j][2])&&(bmax>=complex[j][1])))

{

mmax=complex[j][2];

bmax=complex[j][1];

amax=complex[j][0];

p=j;

}

}

for(k=p;k<size-1;k++)

{

complex[k][0]=complex[k+1][0];

complex[k][1]=complex[k+1][1];

complex[k][2]=complex[k+1][2];

}

size--;

printf("%d+i%d\n",amax,bmax);

printf("SIZE = %d\n",size);

}

}

else

{

scanf("%d+i%d",&x,&y);

complex[size][0]=x;

complex[size][1]=y;

complex[size][2]=x\*x+y\*y;

size++;

printf("SIZE = %d\n",size);

}

}

return 0;

}

20.D4201 &nbsp;&nbsp;&nbsp;输入整数m，将所有大于1小于整数m的素数存入所指定的数组中

#include <stdio.h>

int main(void)

{

int m, cnt = 0, i, j;

int array[101];

printf("Input:\nPlease input a number: \n");

scanf("%d", &m);

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

{

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

{

if (i % j == 0)

{

break;

}

}

if (j == i)

{

array[cnt++] = i;

}

if (cnt > 100)

{

break;

}

}

printf("Output:\n");

if (m <= 0)

{

printf("error!\n");

}

else if (cnt > 100)

{

printf("overflow!\n");

}

else

{

printf("n=%d\n", cnt);

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

{

printf("%5d", array[i]);

if ((i + 1) % 15 == 0)

{

printf("\n");

}

}

printf("\n");

}

return 0;

}

21.HE4304 疯狂立方体！！！

<p>传说中这个立方体中，装有惊天宝藏的秘密，据说将这8个数分别放在立方体的8个顶点</p><p>上，当每个面的四个数之和相等时，立方体就可以解开。

#include<stdio.h>

int main()

{

int a[9],ii=0,i,a1,a2,a3,a4,b1,b2,b3,b4,flag;

printf("Please input 8 number:\n");

for(i=1;i<=8;i++)

{

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

ii+=a[i];

}

printf("Output:\n");

if(ii%2)

{

printf("Sorry they can't be constructed required cube!\n");

return 0;

}

for(flag=0,a1=1;a1<=8;a1++)

for(a2=1;a2<=8;a2++)

if(a2!=a1)

for(a3=1;a3<=8;a3++)

if(a3!=a2 && a3!=a1)

for(a4=1;a4<=8;a4++)

if(a4!=a3 && a4!=a2 && a4!=a1)

for(b1=1;b1<=8;b1++)

if(b1!=a4 && b1!=a3 && b1!=a2 && b1!=a1)

for(b2=1;b2<=8;b2++)

if(b2!=b1 && b2!=a4 && b2!=a3 && b2!=a2 && b2!=a1)

for(b3=1;b3<=8;b3++)

if(b3!=b2 && b3!=b1 && b3!=a4 && b3!=a3 && b3!=a2 && b3!=a1)

for(b4=1;b4<=8;b4++)

if(b4!=b3 && b4!=b2 && b4!=b1 && b4!=a4 && b4!=a3 && b4!=a2 && b4!=a1)

if(a[b1]+a[b2]+a[b3]+a[b4]==ii/2

&&a[a1]+a[a2]+a[b1]+a[b2]==ii/2

&&a[a1]+a[a4]+a[b1]+a[b4]==ii/2)

{

flag=1;

goto out;

}

out:

if(flag)

{

printf("They can be constructed required cube as follow:\n");

printf("\t /%2d--------------/%2d\n",a[a4],a[a3]);

printf("\t%2d/--------------%2d/ |\n",a[a1],a[a2]);

printf("\t | | | |\n");

printf("\t | | | |\n");

printf("\t |%2d| | |%2d\n",a[b4],a[b3]);

printf("\t | /---------------|-/\n");

printf("\t%2d/--------------%2d/\n",a[b1],a[b2]);

}

else

printf("Sorry they can't be constructed required cube!\n");

return 0;

}

22.LB4301 保持数列有序

#include <stdio.h>

int main(void)

{

int i, j, n, x;

int arr[101];

printf("Input:\n");

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

while (n != 0 || x != 0)

{

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

{

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

}

for (j = n - 1; j >= 0 && x < arr[j]; j--)

{

arr[j + 1] = arr[j];

}

arr[j + 1] = x;

printf("Output: ");

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

{

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

}

printf("\n");

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

}

return 0;

}

23.HE4305 ;不差钱儿？？

<p>小明假期同爸爸一起去书店，他选中了六本书，每本书的单价分别为：3.1、1.7、2、5.3、0.9、7.2。

#include<stdio.h>

#include<math.h>

int main()

{

int d[6],m,i,j;

long b[6],flag;

float c[6],min,x;

printf("Please input the prices of 6 books:\n");

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

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

printf("Output:\n");

for(i=0,min=-1,d[0]=0;d[0]<2;d[0]++)

for(d[1]=0;d[1]<2;d[1]++)

for(d[2]=0;d[2]<2;d[2]++)

for(d[3]=0;d[3]<2;d[3]++)

for(d[4]=0;d[4]<2;d[4]++)

for(d[5]=0;d[5]<2;d[5]++)

{

for(flag=0,x=0,j=5;j>=0;j--)

{

x+=c[j]\*d[j];

flag=flag\*10+d[j];

}

x=((x-10>0)?x-10:10-x);

if(min<0)

{

min=x;

b[i++]=flag;

}

else if(min-x>1.e-6)

{

min=x;

b[0]=flag;

i=1;

}

else if(fabs((double)x-min)<1.e-6)

b[i++]=flag;

}

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

{

printf("10(+-)%.2f=",min);

for(flag=b[m],j=0;flag>0;j++,flag/=10)

if(flag%10)

{

if(flag>1)

printf("%.2f+",c[j]);

else

printf("%.2f\n",c[j]);

}

}

}

24.T4203 2064]编排字符串

请输入字符串，要求后输入的字符串排在前面.

#include <stdio.h>

#include <string.h>

int main()

{

int i,j,k,n;

char str[100][20];

printf("Please input n:");

scanf("%d",&n);

getchar();

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

{

gets(str[i]);

}

printf("Output:\n");

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

{

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

{

printf("%d=%s ",j,str[k-1]);

}

printf("\n");

}

}

25.LB4209 给定一个日期，输出这个日期是该年的第几天（多个日期）

描述：给定一个日期，输出这个日期是该年的第几天。

#include <stdio.h>

#include <stdlib.h>

int calculate(int year, int month, int day);

int main(void)

{

int i, n;

int year, month, day;

int \*p = NULL;

printf("Input:\n");

scanf("%d", &n);

p = (int \*)malloc(n \* sizeof(int));

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

{

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

p[i] = calculate(year, month, day);

}

printf("Output:\n");

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

{

printf("%d\n", p[i]);

}

return 0;

}

int calculate(int year, int month, int day)

{

int sum = 0;

switch(month)

{

case 12:

sum += 30;

case 11:

sum += 31;

case 10:

sum += 30;

case 9:

sum += 31;

case 8:

sum += 31;

case 7:

sum += 30;

case 6:

sum += 31;

case 5:

sum += 30;

case 4:

sum += 31;

case 3:

sum += 28;

case 2:

sum += 31;

}

sum += day;

if (((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) && month > 2)

{

sum++;

}

return sum;

}

26.LB4205 对从键盘输入的一行字符(约定：字符数≤127字节)进行排序(按每个字符的ASCII码由小到大)并输出。

#include <stdio.h>

int main(void)

{

int i = 0, j, ch, len;

char te;

char str[128];

printf("input the string: \n");

ch = getchar();

while (ch != '\n' && ch != EOF)

{

str[i] = (char)ch;

i++;

ch = getchar();

}

str[i] = '\0';

len = i;

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

{

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

{

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

{

te = str[j];

str[j] = str[j + 1];

str[j + 1] = te;

}

}

}

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

return 0;

27.HE4301幸运儿

<p>n&nbsp;个人围成一圈,&nbsp;并按顺时针依次编号1～n。从编号为1&nbsp;的人开始，按顺时针方向每隔一人选出一个，剩下的人重新围成一圈，如此循环直到剩下两人，这剩下的两人就是幸运儿。

#include<stdio.h>

int main()

{

int num,i=1,x,n,a[100]={0};

do

{

printf("Please input num:\n");

scanf("%d",&num);

if(num<3 || num>50)

printf("Input error!\n");

}while(num<3 || num>50);

printf("Output:\n");

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

{

a[n]=n+1;

}

x=0;

while(n>=1)

{

if(a[i]!=0)

{

if(x%2==0)

{

if(n==2)

printf("\n");

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

n--;

a[i]=0;

}

x++;

}

if(i+1>=num)

i=0;

else

i++;

}

putchar(10);

return 0;

}

28.HH4209 手机话费

小明的手机每天消费1元，每消费K元就可以获赠1元，一开始小明有M元，问最多可以用多少天？

#include<stdio.h>

int main()

{

int m,k,d;

printf("Input:\n");

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

d=0;

while(m)

{

m--;

d++;

if(d%k==0)

m++;

}

printf("Output:\n%d\n",d);

return 0;

}

29.HH4301 &nbsp;&nbsp;&nbsp;字符串处理

<div><span>读入两个字符串，字符串除了数字还可能包括相加之后输出结果，如果是浮点型，要求用科学计数法表示（最多包含6个有效数字）。

#include<stdio.h>

#include<stdlib.h>

char s1[32],s2[32];

long a1,a2;

int e1,e2;

int main() {

int i,f,s;

long t;

scanf("%s %s",s1,s2);

printf("Output:");

a1=a2=0;

for(s=f=e1=i=0;s1[i];i++) {

if (s1[i]=='-') s=1;

else if (s1[i]=='.') f=1;

else if (s1[i]=='e' || s1[i]=='E') {

e1+=atoi(s1+i+1);

break;

} else a1=a1\*10+s1[i]-'0',e1-=f;

}

if (s) a1=-a1;

for(s=f=e2=i=0;s2[i];i++) {

if (s2[i]=='-') s=1;

else if (s2[i]=='.') f=1;

else if (s2[i]=='e' || s2[i]=='E') {

e2+=atoi(s2+i+1);

break;

} else a2=a2\*10+s2[i]-'0',e2-=f;

}

if (s) a2=-a2;

if (e1<e2) for(;e1<e2;e2--) a2\*=10;

else if (e1>e2) for(;e1>e2;e1--) a1\*=10;

a1+=a2;

if (!a1) {

puts("0");

}

while(e1<0 && a1%10==0) a1/=10,e1++;

if (e1>=0) {

printf("%lld",a1);

for(i=0;i<e1;i++) putchar('0');

puts("");

}

if (a1<0) a1=-a1,s=1; else s=0;

for(t=1,f=0;t<=a1;t\*=10,f++);

if (t>1) t/=10,f--;

if (s) putchar('-');

printf("%lld",a1/t);

if (t>1) printf(".%lld",a1%t);

printf("e%d\n",e1+f);

return 0;

}

30.HH4210 输出最长单词

<p>编写一个函数，输入一行字符，将此字符串中最长的单词输出。

#include <stdio.h>

#include <string.h>

int main() {

void calc(char str[], char longest[]);

char str[101], longest[101];

printf("Input:\n");

gets(str);

calc(str, longest);

printf("Output:\n");

puts(longest);

return 0;

}

void calc(char str[], char longest[]) {

char temp[101];

int l = 0, ret = 0;

while (sscanf(str + l, "%s", temp) != EOF) {

if (strlen(temp) > ret) {

ret = strlen(temp);

strcpy(longest, temp);

}

l += strlen(temp);

}

}

31.LB4101 求任意的一个m×m矩阵(2<=m<=20)的对角线上元素之和

#include <stdio.h>

int main(void)

{

int i, j, m, sum = 0;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

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

{

sum = sum + array[i][i] + array[i][m - i - 1];

}

if (m % 2 == 1)

{

sum = sum - array[m / 2][m /2];

}

printf("Output:\nsum=%d\n", sum);

return 0;

}

32.WB4205 找一串字符串有多少个整数

#include <stdio.h>

int main() {

int i, current, count, vals[500];

char ch, lastch;

printf("请输入一串可带有任意字符的字符串:\n");

count = 0;

/\* 使用current变量标志当前是否有没有处理完毕的整数

-1表示没有，否则表示当前整数已处理的值 \*/

current = -1;

while ((ch = getchar()) != '\n') {

if ('0' <= ch && ch <= '9') {

if (current == -1) {

current = ch - '0';

} else {

current = current \* 10 + (ch - '0');

}

} else {

if (current != -1) {

vals[count] = current;

count++;

current = -1;

}

}

}

/\* 不要忘记在读取完整个字符串之后判断一下是否还有

未处理完毕的整数 \*/

if (current != -1) {

vals[count] = current;

count++;

current = -1;

}

printf("\nOutput:\n");

printf("\n现从你输入的字符串中检测到了%d个整数。\n\n它们分别是:", count);

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

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

printf("\n\n");

return 0;

}

33.LB4207 求任意的一个m×m矩阵(2<=m<=20)的周边元素之和

#include <stdio.h>

int main(void)

{

int i, j, m, sum = 0;

int array[20][20];

printf("Please input m:");

scanf("%d", &m);

printf("\nPlease input array:\n");

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

{

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

{

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

}

}

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

{

if (i == 0 || i == m - 1)

{

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

{

sum += array[i][j];

}

continue;

}

sum = sum + array[i][0] + array[i][m - 1];

}

printf("Output:\nsum=%d\n", sum);

return 0;

}

34.WB4202 职员按编号大小排序并按编号查找某职员

#include <stdio.h>

#include <string.h>

int main() {

void input(char name[][21], int num[]);

void sort(char name[][21], int num[]);

void check(char name[][21], int num[], int target, char output[]);

char name[10][21], output[21];

int num[10];

int i, target;

input(name, num);

sort(name, num);

printf("Please input a number you input before:");

scanf("%d", &target);

printf("\nOutput:\nThe sort list is:\n");

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

printf("%s %d\n", name[i], num[i]);

check(name, num, target, output);

printf("The name you want to find is:");

puts(output);

return 0;

}

void input(char name[][21], int num[]) {

int i;

printf("Please input 10 names and 10 numbers[1-1000]:\n");

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

scanf("%s%d", name[i], &num[i]);

}

void sort(char name[][21], int num[]) {

int i, j, temp;

char tempname[21];

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

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

if (num[i] > num[j]) {

temp = num[i];

num[i] = num[j];

num[j] = temp;

strcpy(tempname, name[i]);

strcpy(name[i], name[j]);

strcpy(name[j], tempname);

}

}

}

}

void check(char name[][21], int num[], int target, char output[]) {

int low, high, mid;

low = 0;

high = 10;

while (low < high) {

mid = (low + high) / 2;

if (num[mid] < target)

low = mid + 1;

else

high = mid;

}

strcpy(output, name[low]);

}

35.T4202 [2062]姓名排序

存储一组姓名，如Apple,Tom,Green,Jack;要求按照字典序排序并显示。

#include <stdio.h>

#include <string.h>

int main()

{

int n,i,j;

char str[10][20];

char string[20];

printf("Please input n:");

scanf("%d",&n);

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

scanf("%s",&str[i]);

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

{

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

{

if(strcmp(str[j],str[j+1])>0)

{

strcpy(string,str[j]);

strcpy(str[j],str[j+1]);

strcpy(str[j+1],string);

}

}

}

printf("Output:\n");

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

printf("%s\n",str[i]);

}

&nbsp;36.LB4206 有一递推数列，满足f(0)=0,f(1)=1,f(2)=2, f(n+1)=2f(n)+f(n-1)f(n-2) (n>=2), 编写程序求f(n)的值

#include <stdio.h>

int main(void)

{

int i, n;

double array[21];

printf("Input n (13>=n>=2):");

scanf("%d", &n);

array[0] = 0;

array[1] = 1;

array[2] = 2;

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

{

array[i] = 2 \* array[i - 1] + array[i - 2] \* array[i - 3];

}

printf("Output:\nf(%d)=%.0f\n", n, array[n]);

return 0;

}

37.HE4203 找完数

#include<stdio.h>

int main()

{

int k[100];

int i,j,n,num,s;

printf("Please input num: ");

scanf("%d",&num);

printf("Output:\n");

if(num<=0 || num>10000)

{

printf("Input error!\n");

return 0;

}

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

{

n=-1;

s=j;

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

{

if((j%i)==0)

{

n++;

s=s-i;

k[n]=i;

}

}

if(s==0)

{

printf("%d is a wanshu: ",j);

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

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

printf("%d\n",k[n]);

}

}

return 0;

}

38.LB4105 求斐波拉契数列的前n（n的范围为：3～20）项

#include <stdio.h>

int main(void)

{

int i, n;

int array[21] = {1, 1};

printf("input a data(3--20):");

scanf("%d", &n);

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

{

array[i] = array[i - 1] + array[i - 2];

}

printf("Output:\n");

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

{

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

}

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

return 0;

}

39.LB4103从键盘输入一行可带空格的字符串(约定：字符数≤127字节)，按逆序输出该字符串

#include <stdio.h>

#include <string.h>

int main(void)

{

int i;

char string[128];

printf("Input a string: ");

gets(string);

printf("Output: ");

for (i = strlen(string) - 1; i >= 0; i--)

{

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

}

printf("\n");

return 0;

}

40.LB4102 输入3行3列的矩阵，输出所有元素的累加和。

#include <stdio.h>

int main(void)

{

int i, j, sum = 0;

int array[3][3];

printf("Please input the 3x3 Matrix:\n");

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

{

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

{

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

sum += array[i][j];

}

}

printf("Output:\nsum=%d\n", sum);

return 0;

}

41.HE4303 拉丁方阵

<p>构造N\*N阶的拉丁方阵，使方阵中的每一行和每一列中的数字1到N只出现

#include<stdio.h>

#define N 9

int main()

{

int j,i,k,t,n;

int DATA[N][N][N];

printf("Please input N: ");

scanf("%d",&n);

printf("Output:\n");

if(n<=0 || n>6)

{

printf("Input error!\n");

return 0;

}

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

{

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

{

t=(i+j)%n;

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

DATA[j][i][k]=(k+t)%n+1;

}

}

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

{

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

{

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

{

printf("%2d",DATA[i][j][k]);

}

printf(" ");

}

printf("\n");

}

return 0;

}