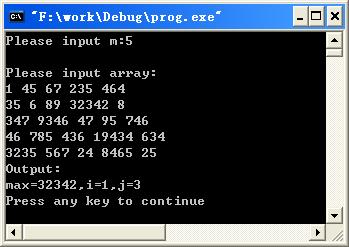
LB4202求任意的一个m×m矩阵的最大数及其所在的行列数，m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型)。

程序的运行效果应类似地如图所示，图中的5和  
1 45 67 235 464  
35 6 89 32342 8   
347 9346 47 95 746  
46 785 436 19434 634  
3235 567 24 8465 25  
是从键盘输入的内容。



#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;

}

}

}

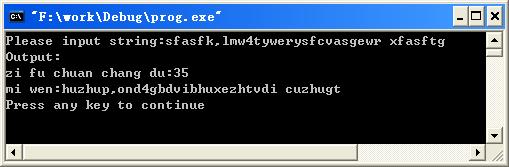
printf("Output:");

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

return 0;

}

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：  
　　　　A->Z   　a->z  
　　　　B->Y   　b->y  
　　　　C->X   　c->x  
　　　　……   　……  
　　　　Z->A   　z->a  
　　即字母A变成Z、字母B变成Y...，非字母字符不变。最后在屏幕上先显示这一行字符的长度，再显示生成的密文。  
　　编程可用素材：printf("Please input string:")...、printf("Output:\nzi fu chuan chang du:...、printf("\nmi wen:...。  
　　程序的运行效果应类似地如图1所示，图1中的sfasfk,lmw4tywerysfcvasgewr xfasftg是从键盘输入的内容



#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');

}

}

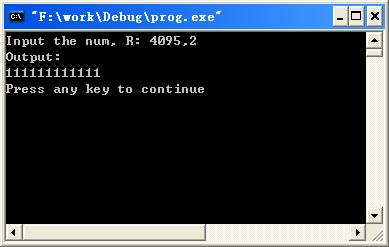
printf("Output:\nzi fu chuan chang du:%d", i);

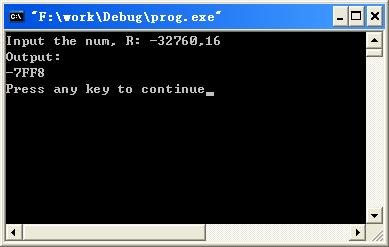
printf("\nmi wen:%s\n", string);

return 0;

}

LB4302实现十进制转num换成R进制数，num和R从键盘读入(假定num为int且-32767<=num<=32767，R为int且2<=R<=16)。编程可用素材：printf("Input the num, R: ")...、printf("Output:\n")...。  
　　程序的运行效果应类似地如图1和图2所示，图1中的4095,2和图2中的-32760,16是从键盘输入的内容。





#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--)

{

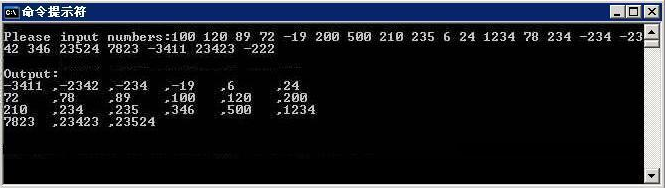
str[i] = str2[j];

}

str[i] = '\0';

}

LB4201①程序运行时先显示Please input numbers:，再从键盘上读入一组整数(只考虑int型)，数与数之间只使用空格或回车作分隔。数可正可负，最多10000个，但若读入的数为-222时，则表示输入结束且-222不算在该组数内。  
　　②对这一组数按从小到大的顺序进行排序。  
　　③将排序后的这一组数输出到屏幕上，输出格式为每行6个数，数与数之间使用逗号(,)分隔，两个逗号之间的宽度(不算逗号)为6且使用左对齐格式。注意，行尾没有逗号。



#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");

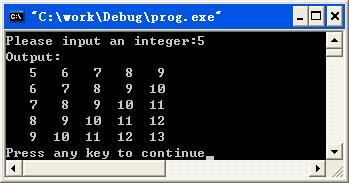
}

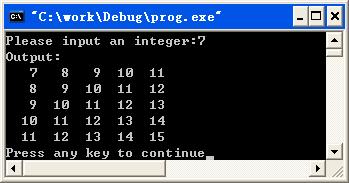
}

return 0;

}

LB4104编写程序实现以下功能  
　　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。  
　　程序的运行效果应类似地如图1和图2所示，图1中的5和图2中的7是从键盘输入的内容。





#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]);

}

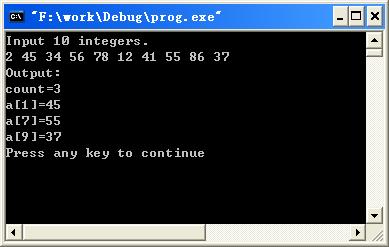
printf("\n");

}

return 0;

}

LB4202输入10个整型数存入一维数组，输出值和下标都为奇数(数组第1个元素的下标为0)的元素及其个数，要求先输出个数。编程可用素材：printf("Input 10 integers.\n")...、printf("Output:\ncount=...\n"...、printf("a[...]=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的2 45 34 56 78 12 41 55 86 37是从键盘输入的内容。



#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++)

{

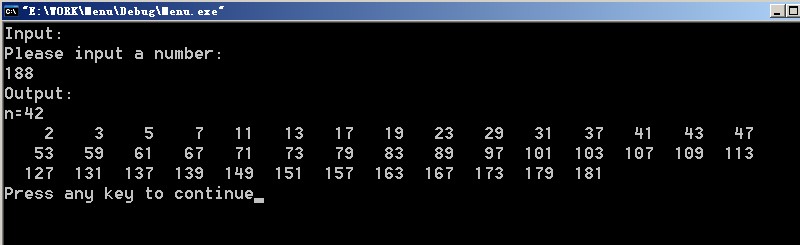
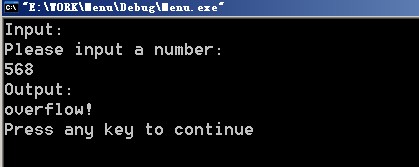
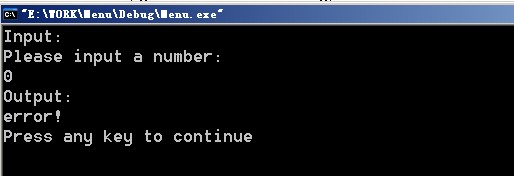
printf("a[%d]=%d\n", barray[i], array[barray[i] ]);

}

return 0;

}

D4201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(数组最多只存放100个素数，超过则提示“overflow”)，输出素数的个数n及各素数——素数的输出格式为每个素数5列宽、右对齐、每行显示15个。若输入的m≤0，则提示“error”，程序终止。注：素数(Prime Number)，亦称质数，指在一个大于1的自然数中，除了1和此整数自身外，没法被其他自然数整除的数。  
　　编程可用素材：printf("input a number: ")...、printf("error")...、printf("overflow")...、printf("n=...\n"...。  
　　程序的运行效果应类似地如图1、图2和图3所示，图1中的188、图2中的0和图3中的568是从键盘输入的内容



#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");

}

}

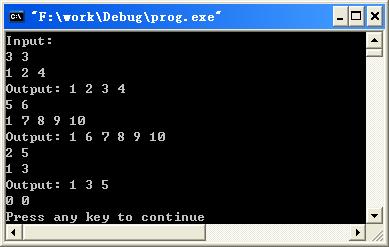
printf("\n");

}

return 0;

}

LB4301保持数列有序：有n(约定n<=100)个整数，已经按照从小到大顺序排列好，现在另外给一个整数x，请将该数插入到序列中，并使新的序列仍然有序。  
　　程序输入：输入数据包含多组测试实例，每组数据由两行组成，第一行是n和x，第二行是已经有序的n个数的数列。n和x同时为0表示输入数据的结束，本行不做处理。  
　　程序输出：对于每个测试实例，输出插入新的元素后的数列。  
　　程序的运行效果应类似地如图1所示，图1中的3 3、1 2 4和0 0是从键盘输入内容。



#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;

}