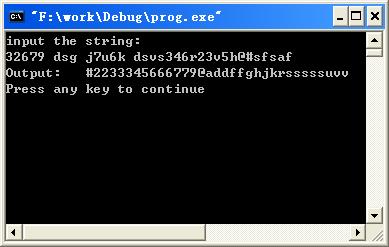
LB4205对从键盘输入的一行字符(约定：字符数≤127字节)进行排序(按每个字符的ASCII码由小到大)并输出。编程可用素材：printf("input the string: \n")...、printf("Output:")...。  
程序的运行效果应类似地如图1所示，图1中的32679 dsg j7u6k dsvs346r23v5h@#sfsaf是从键盘输入的内容。



#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;

}

}

}

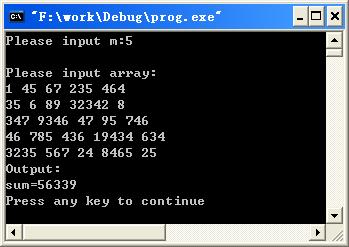
printf("Output:%s\n", str);

return 0;

}

LB4101求任意的一个m×m矩阵的主副对角线上元素之和（注：每个元素只计一次），m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型，且不需考虑求和的结果可能超出int型能表示的范围)。

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, 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];

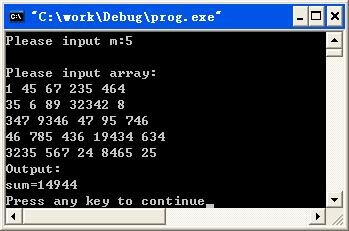
}

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

return 0;

}

LB4207编写一程序实现以下功能  
　　求任意的一个m×m矩阵的周边元素之和，m(2<=m<=20)及矩阵元素从键盘输入(只考虑int型，且不需考虑求和的结果可能超出int型能表示的范围)。编程可用素材：printf("\nPlease input m: ")...、printf("\nPlease input array:\n")...、printf("Output:\nsum=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1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。



/\*

5

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

\*/

#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];

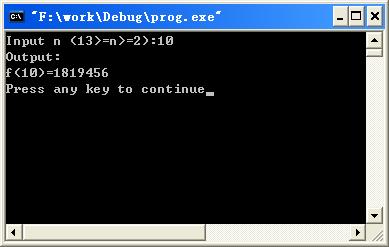
}

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

return 0;

}

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（n由键盘输入，13>=n>=2）。编程可用素材：printf("Input n (13>=n>=2): ")...、printf("Output:\nf(...)=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的10是从键盘输入的内容。



#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;

}

LB4105求斐波拉契数列的前n（n的范围为：3～20）项，并输出到屏幕上（数和数之间用字符'\t'隔开）。斐波拉契公式：f(1)=1,f(2)=1,f(n)=f(n-1)+f(n-2) (n≥3)。编程可用素材：printf("input a data(3--20):")...printf("Output:\n")...。  
　　程序的运行效果应类似地如图1所示，图1中的11是从键盘输入的内容。



#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]);

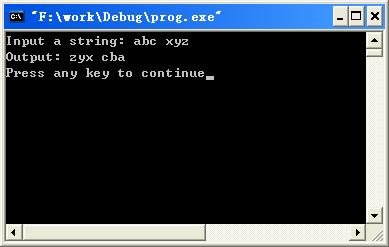
}

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

return 0;

}

LB4103编写程序实现以下功能  
　　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。编程可用素材：printf("Input a string: ")...、printf("Output: ")...。  
　　程序的运行效果应类似地如图1所示，图1中的abc xyz是从键盘输入的内容。



#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]);

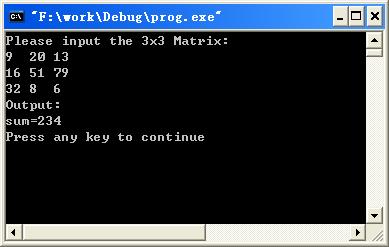
}

printf("\n");

return 0;

}

LB4102输入3行3列的矩阵，输出所有元素的累加和（整型）。编程可用素材：printf("Please input the 3x3 Matrix:\n"...、printf("Output:\nsum=...\n"...。  
　　程序的运行效果应类似地如图1所示，图1中的  
        9  20 13  
        16 51 79  
        32 8  6  
　　是从键盘输入的内容。



#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;

}