w10 集合基本运算

1. 运行结果：



1. 代码改进与注释（红字为此段伪代码解释，红字加下划线为修改过的代码解释，//黑字为单句解释）

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

//集合运算

void printYsh(char a[])

{

int nLen=0;

int i=0;

nLen=strlen(a);

if(nLen==0)

{printf("空集");

}

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

{

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

if(i!=(nLen-1))

{

printf(",");

}

}

printf("\n");

}

首先，如果是空集情况，就打印下来，不是空集的话，建立for循环，for循环内增加判断，如果i不等于nlen-1再打印“，”，使最后一个数打印完之后不用加没有必要的“，”。并且，把原本的int型返回值的函数改成void。

void printYsh2(char a[][3],int nLen)

{

int i=0;

if(nLen==0)

{printf("空集");

}

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

{

printf("<%c,%c>",a[i][0],a[i][1]);

if(i!=(nLen-1))

{

printf(",");

}

}

printf("\n");

}

增加一个函数，考虑到<a,b>这样的情况。

int inputYsh(char a[])

{

int nLen=0,i=0,j=0,k=0;

char stmp[1024];

printf("集合元素只能是A-Z,a-z,0-9,其他字符被当做分隔符去掉;\n");

gets(stmp);

fflush(stdin);

nLen=strlen(stmp);

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

{

if(((stmp[i]>='A')&&(stmp[i]<='Z'))||((stmp[i]>='a')&&(stmp[i]<='z'))||((stmp[i]>='0')&&(stmp[i]<='9')))

{

//是否重复

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

{

if(a[k]==stmp[i])

{break;}

}

if(k>=j)

{

a[j]=stmp[i];

j++;

}

}

}

a[j]='\0';

return j;

}

int set(char a[],char b[],char c[],char d[],char e[],char f[],char g[][3],char h[][3])

{

int i=0,j=0,k=0,l=0,xx=0,m=0,n=0,o=1,gg=0,hh=0,nLen1=0,nLen2=0;

nLen1=strlen(a);

nLen2=strlen(b);

l=nLen1;

for(i=0;i<nLen1;i++){d[i]=a[i];}

for(j=0;j<nLen2;j++)

{

for(xx=0;xx<l;xx++)

{

if(d[xx]==b[j]){break;}

}

if(xx>=l){d[l]=b[j];l++;}

}

for(i=0;i<nLen1;i++)

{

o=1;

for(j=0;j<nLen2;j++)

{

g[gg][0]=a[i];

g[gg][1]=b[j];

g[gg][2]='\0';

gg++;

h[hh][0]=b[j];

h[hh][1]=a[i];

h[hh][2]='\0';

hh++;

if(a[i]==b[j])

{

o=0;

c[k]=a[i];

k++;

}

}

if(o==1){

e[m]=a[i];

m++;}

}

for(i=0;i<nLen2;i++)

{

o=1;

for(j=0;j<nLen1;j++)

{

if(b[i]==a[j]){o=0;}

}

if(o==1)

{

f[n]=b[i];

n++;

}

}

c[k]='\0';

d[l]='\0';

e[m]='\0';

f[n]='\0';

g[gg][0]='\0';

h[hh][0]='\0';

return gg;

}

我把三个函数放到一起，用char a[],char b[],char c[],char d[],char e[],char f[],char g[][3],char h[][3]来作为变量

g[gg][0]=a[i];

g[gg][1]=b[j];

g[gg][2]='\0';

gg++;

h[hh][0]=b[j];

h[hh][1]=a[i];

h[hh][2]='\0';

hh++；

这里，把g和h的数组进行循环赋值，为之后的ysh2的变量做准备。

if(a[i]==b[j])

{

o=0;

c[k]=a[i];

k++;

}

有相等的，并集+1

没有的话，if(o==1){

e[m]=a[i];

m++;}

int main()

{

char a[1024],b[1024],c[1024],d[1024],e[1024],f[1024],g[1024][3],h[1024][3];

int nLen=0;

int i=0;

printf("输入集合a:");

inputYsh(a);

printf("输入集合b:");

inputYsh(b);

printf("A:");

printYsh(a);

printf("B:");

printYsh(b);

i=set(a,b,c,d,e,f,g,h);

printf("交集:");

printYsh(c);

printf("并集:");

printYsh(d);

printf("A-B:");

printYsh(e);

printf("B-A:");

printYsh(f);

printf("A\*B:");

printYsh2(g,i);

printf("B\*A:");

printYsh2(h,i);

return 0;

}

a\*b和b\*a用之前设定的ysh2来输出就可以。