第二题

#include <stdio.h>

#include <string.h>

int main()

{

int res[80];//存放纯数字

int ans[10];

char str[80];//原数组

int p,i,j,n,total,len;

total=0;

p=0;

gets(str);

len=strlen(str);

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

{

if(str[i]>='0'&&str[i]<='9')

{

res[p]=str[i]-'0';

p++; //p=数组中最后一个元素下标+1

}

}

for(i=0;i<=9;i++) //找数字0~9

{

n=0;

for(j=0;j<p;j++) //从数组里面一个个找

{

if(res[j]==i) //符合

{

n++;

total++;

}

}

ans[i]=n; //储存于数组

}

for(i=0;i<=9;i++)

{

if(ans[i]!=0)

{

printf("%d->=%d,",i,ans[i]);

}

}

printf("TOTAL->%d",total);

return 0;

}

我的做法

#include <stdio.h>

#include <string.h>

//思路：1.把文本中的数字按文本顺序提出来 2.计算数组中数字的个数

int main()

{

char str[80],res[80];

int ans[10],i,j,len,total=0,p=0;

gets(str);

len=strlen(str);

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

{

if(str[i]>='0'&&str[i]<='9') //把文本中的数字按文本顺序提进res

{

res[p]=str[i];

p++;

}

}

for(i=0;i<=9;i++) //计算数组中数字的个数

{

int sum=0; //注意这个sum放置的位置

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

{

if(i==res[j]-'0')

{

sum++;

total++;

}

}

ans[i]=sum;

}

for(i=0;i<=9;i++)

{

if(ans[i]!=0)

{

printf("%d->%d,",i,ans[i]);

}

}

printf("TOTAL->%d",total);

return 0;

}

第三题

#include <stdio.h>

#include <string.h>

void yasuo(char str[],char ans[])

{

int p=1,i,j,len,flag;

len=strlen(str);

ans[0]=str[0];

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

{

if((str[i]>='a'&&str[i]<='z')||(str[i]>='A'&&str[i]<='Z')) //字母才能保留

{

flag=0;

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

{

if(str[i]==str[j]) flag++; //判断有没有重复两次

}

if(flag<2) //没有重复两次（符合保留标准）

{

ans[p]=str[i];

p++;

}

}

}

}

int main()

{

char str[50],ans[50];

int len,i;

gets(str);

yasuo(str,ans);

len=strlen(ans);

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

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

return 0;

}

第四题

/\*int subst(int \* a, int m, int \* b ,int n,int \* c);

函数中a、b和c为数组，m和n代表a和b的元素个数，返回c的元素个数。\*/

#include <stdio.h>

int subst(int \* a, int m, int \* b ,int n,int \* c)

{

int i,j,k;

int flag,p=0;

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

{

flag=0; //小心flag的位置在循环内

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

{

if(a[i]==b[j]) flag=1; //flag=1即不满足条件 ,等于符号是两个等号！！！

}

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

{

if(a[i]==c[k]) flag=1;

}

if(flag==0)

{

c[p]=a[i];

p++;

}

}

return p;

}

/\*在主函数中输入正整数m和n，再输入a和b各元素的值，然后调用函数subst，

最后在主函数显示c的全部元素值（顺序为它们在a中的顺序）。如果差集为空，输出 None\*/

int main()

{

int \*x,\*y,\*z,a[100],b[100],c[100];

int m,n,i,j,s;

x=a,y=b,z=c;

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

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

{

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

}

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

{

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

}

s=subst(a,m,b,n,c);

if(s==0) printf("None");

else

{

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

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

}

return 0;

}

第五题

/\*（1）定义应聘人员信息的结构体CANDIDATE，

包括姓名（由不含空格的26个英文字母的大小写构成，不超过10个字符）、年龄、

5位面试官打分和总分，其中打分和总分均为浮点数；\*/

#include <stdio.h>

struct CANDIDATE

{

char name[10];

int age;

float s1,s2,s3,s4,s5,all;

}candi[5];

/\*（2）定义void input(struct CANDIDATE \*candi)函数，

用来输入一位应聘人员的信息（姓名和年龄）和面试得分，计算总分并保存；\*/

void input(struct CANDIDATE \*candi)

{

scanf("%s%d%f%f%f%f%f",&candi->name,&candi->age,&candi->s1,&candi->s2,&candi->s3,&candi->s4,&candi->s5);

candi->all=candi->s1+candi->s2+candi->s3+candi->s4+candi->s5;

}

/\*（3）定义void output(struct CANDIDATE \*candi)函数，

用来输出一位应聘人员的信息（姓名和年龄）和面试总分（总分保留2位小数）；\*/

void output(struct CANDIDATE \*candi)

{

printf("%s %d %.2f",candi->name,candi->age,candi->all);

}

/\*（4）定义void select(struct CANDIDATE \*candi)函数，

用来选出总分最低的前2名（假设不存在总分相同的情况）；

提示：可排序；注意：只需要选出前2名。\*/

void select(struct CANDIDATE \*candi)

{

int i,j,k;

struct CANDIDATE temp;

for(i=0;i<2;i++)

{

k=i;

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

{

if(candi[i].all>candi[j].all) k=j;

}

if(k!=i)

{

temp=candi[i];

candi[i]=candi[k];

candi[k]=temp;

}

}

}

int main()

{

int i;

for(i=0;i<5;i++) input(&candi[i]);

select(candi);

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

{

printf("Number%d: ",i+1);

output(&candi[i]);

printf("\n");

}

return 0;

}