1:计算字符串最后一个单词的长度，单词以空格隔开，字符串长度小于5000。

输入：hello nowcoder

输出：8最后一个单词为nowcoder，长度为8

int main()

{

char str[5000]={0};

int len=0;

int i=0;

int count=0;

char \*p=NULL;

fgets(str,5000,stdin);

if((p = strchr(str,'\n')) != NULL)

\*p = '\0';//手动将\n位置处的值变为0

len = strlen(str);

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

{

if(str[i]!=' '&&str[i]!='\0')

{

count++;

}

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

{

i=-1;

}

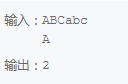
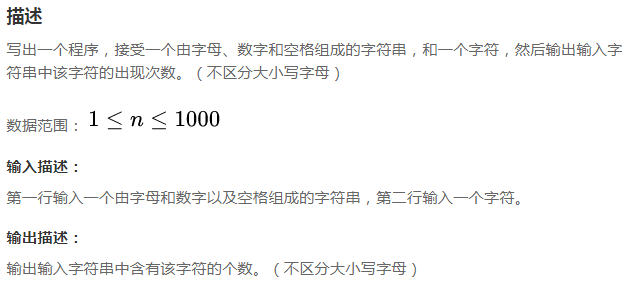
}

printf("%d\n",count);

return 0;

}

**计算某字符出现次数**



int main()

{

char c[5000]={0};

char cc;

int i=0;

int cnt=0;

while((cc=getchar())!='\n')

{

c[i]=cc;

i++;

}

cc=getchar();

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

{

if(cc>='0'&&cc<='9'||cc==' ')

{

if(c[j]==cc)

{

cnt++;

}

}

else

{

if(c[j]==cc||c[j]+32==cc||c[j]-32==cc)

{

cnt++;

}

}

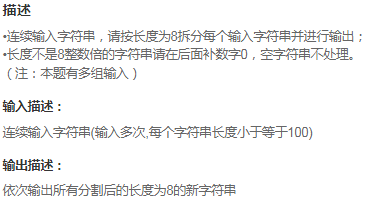
}

printf("%d",cnt);

return 0;

}

**字符串分隔**



int main()

{

char str[100];

int len=0;

int i=0;

int need =0;

while(gets(str))

{

len = strlen(str);

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

{

if(0==(i+1)%8)

{

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

}

else

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

}

if(len%8!=0)

{

need=8-len%8;

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

{

printf("0");

}

printf("\n");

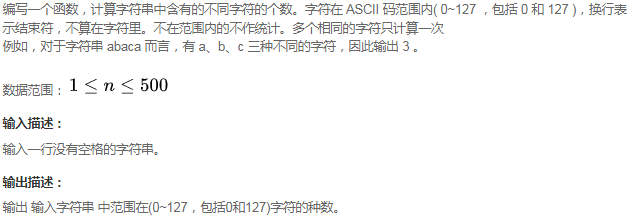
}

}

return 0;

}

**字符个数统计:**



int main()

{

int scc[128]={0};

char c;

int n;

int sum=0;

while(scanf("%c",&c)!=EOF)

{

n=(int)c;

if(n>=0&&n<=127&&scc[n]==0&&n!=10)

{

sum++;

scc[n]=1;

}

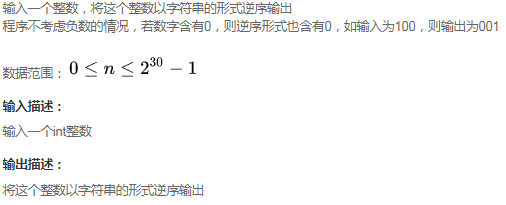
}

printf("%d",sum);

return 0;

}

数字颠倒：

int main()

{

char ch[50];

int i=0;

int j=0;

int n=0;

int d=0;

scanf("%d",&n);

while(n%10)

{

d= n%10;

ch[i]=char(d+'0');

i++;

n=n/10;

}

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

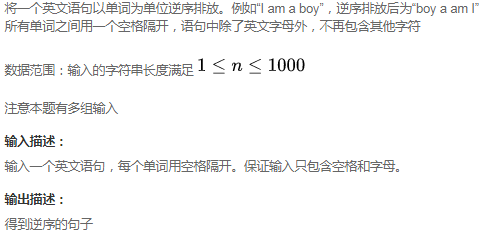
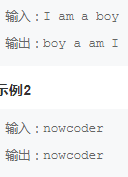
{

printf("%c",ch[j]);

}

}

**句子逆序：**

int main()

{

char str[1000];

gets(str);

int i=strlen(str)-1;

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

{

if(str[i]==' ')

{

for(int j=i+1;str[j]!=' '&&str[j]!='\n'&&str[j]!='\0';j++)

{

printf("%c",str[j]);

}

printf(" ");

}

}

for(i=0;str[i]!=' '&&str[i]!='\n'&&str[i]!='\0';i++)

{

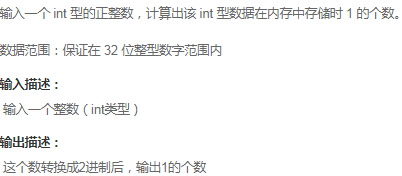
printf("%c",str[i]);

}

return 0;

}

**求int型正整数在内存中存储时1的个数**



**int main()**

**{**

**int res=0;**

**int num=0;**

**int n=0;**

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

**while(n>0)**

**{**

**res=n%2;**

**if(res==1)**

**{**

**num++;**

**}**

**n=n/2;**

**}**

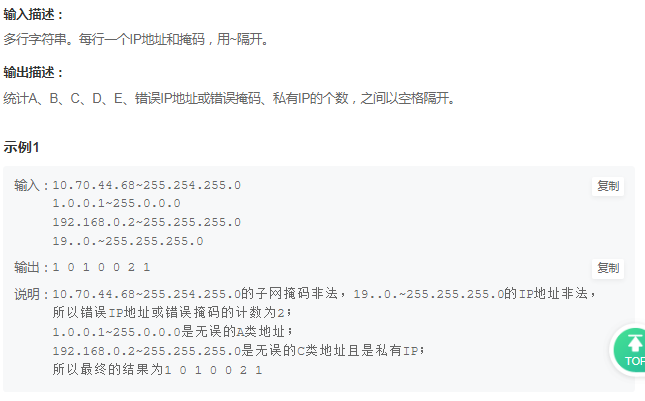
**printf("%d\n",num);**

**return 0;**

**}**

**识别有效的IP地址和掩码并进行分类统计：**





**#include <stdio.h>**

**#include <arpa/inet.h>**

**#include <sys/socket.h>**

**#include <netinet/in.h>**

**#include <strings.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#define true 1**

**#define false 0**

**int validMask(char \*p)**

**{**

**int flag,i ;**

**unsigned int b1 = 0, n[4];**

**sscanf(p, "%u.%u.%u.%u", &n[3], &n[2], &n[1], &n[0]);**

**if(n[0] == 255 &&n[1] == 255 &&n[2] == 255 &&n[3] == 255 )**

**{**

**flag = false;**

**return flag;**

**}**

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

**b1 += n[i] << (i \* 8);**

**b1 = ~b1 + 1; //合法的子网掩码，按位取反，加一后，二进制位中，只有一个1**

**if((b1 & (b1 - 1)) == 0) // 与减1相与，若为0，表示只有某一位为1**

**{**

**flag = true;**

**}**

**else**

**flag = false;**

**return flag;**

**}**

**int main()**

**{**

**char str[50];**

**int a =0, b = 0, c = 0, d = 0, e = 0, err = 0, pri = 0;**

**//while(fgets(str,50,stdin))**

**while(gets(str))**

**{**

**char \*tok = str;**

**char p[2][20] = {0};**

**int i = 0;**

**while ((tok = strtok(tok, "~")) != NULL)**

**{**

**strcpy(p[i], tok);**

**tok = NULL;**

**i ++ ;**

**if(i == 2)**

**i = 0;**

**}**

**int flag = validMask(p[1]);**

**if(flag)**

**{**

**struct in\_addr s;**

**unsigned int ip1,ip2,ip3,ip4;**

**int valid = inet\_pton(AF\_INET,p[0],(void \*)&s);**

**sscanf(p[0],"%u.%u.%u.%u",&ip1,&ip2,&ip3,&ip4);**

**if(valid)**

**{**

**if(ip1>=1 && ip1 <=126)**

**a++;**

**else if(ip1>=128 && ip1 <=191)**

**b++;**

**else if(ip1>=192 && ip1 <=223)**

**c++;**

**else if(ip1>=224 && ip1 <=239)**

**d++;**

**else if(ip1>=240 && ip1 <=255)**

**e++;**

**if(ip1==10**

**|| (ip1==172 && ip2 >=16 &&ip2 <=31)**

**|| (ip1==192 && ip2 ==168))**

**pri ++;**

**}**

**else**

**err ++;**

**}**

**else**

**err++;**

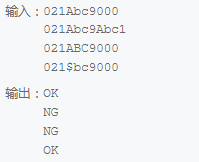
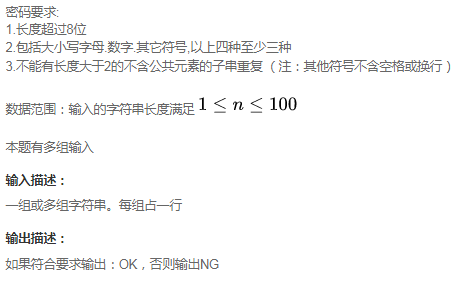
**}**

**printf("%d %d %d %d %d %d %d\n",a,b,c,d,e,err,pri);**

**return 0;**

**}**

**密码验证合格程序：**



**int check\_psw\_type\_vaild (char \*psw)**

**{**

**int low\_flag=0;**

**int big\_flag=0;**

**int num\_flag=0;**

**int alp\_flag=0;**

**if(NULL==psw)**

**{**

**printf("param is illegal \n");**

**return -1;**

**}**

**for(int i=0;i<strlen(psw);i++)**

**{**

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

**{**

**low\_flag=1;**

**}**

**else if(psw[i]>='A'&&psw[i]<='Z')**

**{**

**big\_flag=1;**

**}**

**else if(psw[i]>='0'&&psw[i]<='9')**

**{**

**num\_flag=1;**

**}**

**else**

**{**

**alp\_flag=1;**

**}**

**if(low\_flag+big\_flag+num\_flag+alp\_flag>=3)**

**{**

**return 1;**

**}**

**}**

**return 0;**

**}**

**int main()**

**{**

**char \*str =(char \*)malloc(sizeof(char)\*50);**

**while(gets(str))**

**{**

**char \*pstr=str;**

**int len=strlen(str);**

**if(len<=8)**

**{**

**printf("NG\n");**

**continue;**

**}**

**if(1!=check\_psw\_type\_vaild(str))**

**{**

**printf("NG\n");**

**continue;**

**}**

**int flag=0;**

**for(int i=0;i<(len-4);i++)**

**{**

**for(int j=i+1;j<(len-3);j++)**

**{**

**if(strncmp(pstr+i,pstr+j,3)==0)**

**{**

**printf("NG\n");**

**flag=1;**

**break;**

**}**

**}**

**if(flag)**

**{**

**break;**

**}**

**}**

**if(flag)**

**{**

**continue;**

**}**

**printf("OK\n");**

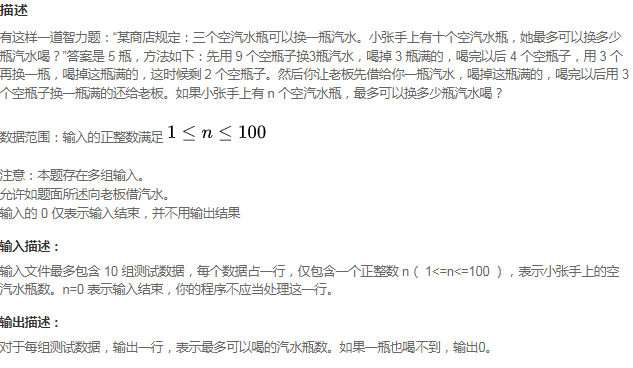
**}**

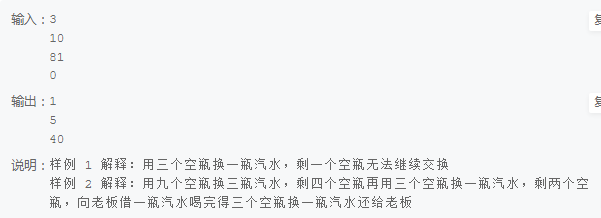
**free(str);**

**return 0;**

**}**

**汽水瓶：**





**int method(int num)**

**{**

**int a=0;**

**int b=0;**

**//printf("%d\n",num);**

**if(num==0||num==1)**

**{**

**return 0;**

**}**

**else if(num==2||num==3)**

**{**

**return 1;**

**}**

**else**

**{**

**a=num/3;**

**b=num%3;**

**return a+method(a+b);**

**}**

**}**

**int main()**

**{**

**int num=0;**

**int count=0;**

**while( scanf("%d",&num)!=EOF)**

**{**

**if(num==0)**

**{**

**break;**

**}**

**//printf("%d\n",num);**

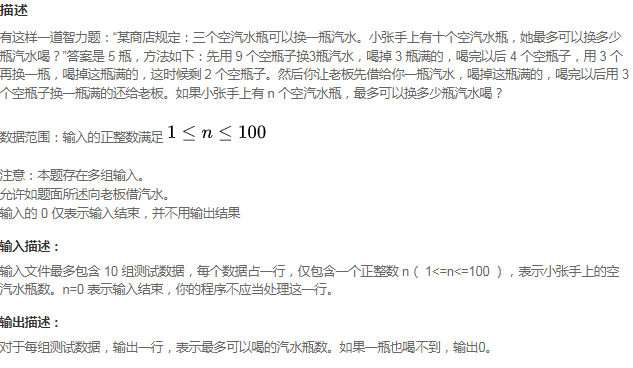
**count = method(num);**

**printf("%d\n",count);**

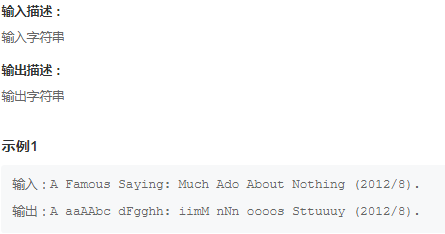
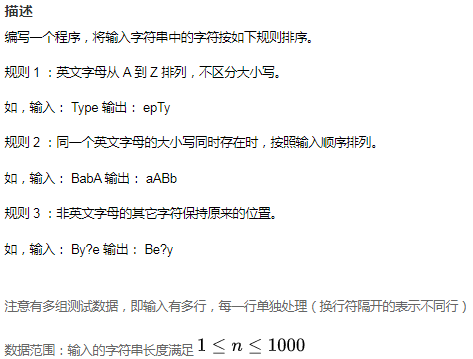
**}**

**return 0;**

**}**



**字符串排序：**



**int judge\_alp(char s)**

**{**

**if(s>='a'&&s<='z')**

**return s-'a'+1;**

**else if(s>='A'&&s<='Z')**

**return s-'A'+1;**

**return 0;**

**}**

**int main()**

**{**

**char str[1000];**

**while(fgets(str,sizeof(str),stdin)!=NULL)**

**{**

**char temp[1000]={0};**

**int len = strlen(str);**

**int i,j,key=0;**

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

**{**

**if(judge\_alp(str[i]))**

**{**

**temp[key]=str[i];**

**for(j=key-1;j>=0;j--)**

**{**

**if(judge\_alp(temp[j])>judge\_alp(str[i]))**

**{**

**temp[j+1]=temp[j];**

**temp[j]=str[i];**

**}**

**}**

**key++;**

**}**

**}**

**j=0;**

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

**{**

**if(judge\_alp(str[i]))**

**{**

**str[i]=temp[j];**

**j++;**

**}**

**}**

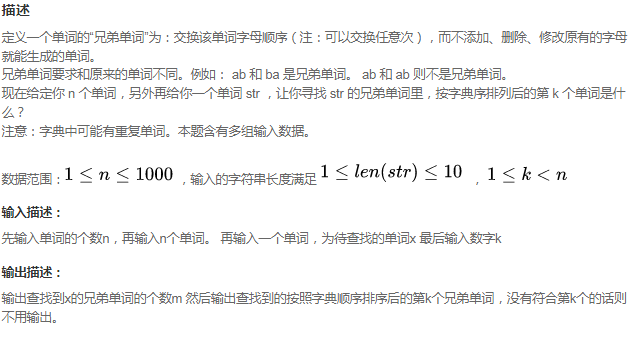
**printf("%s",str);**

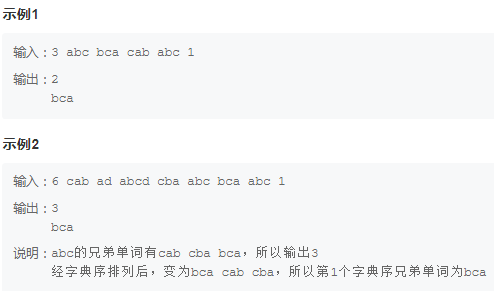
**}**

**return 0;**

**}**

**查找兄弟单词：**





**int judge\_bianwei(char \*a,char\*b)**

**{**

**if(NULL==a||NULL==b)**

**{**

**return 0;**

**}**

**if(0==strcmp(a,b))**

**{**

**return 0;**

**}**

**int alen=strlen(a);**

**int blen=strlen(b);**

**if(alen!=blen)**

**return 0;**

**char c['z'-'A']={0};**

**char d['z'-'A']={0};**

**for(int i=0;i<alen;i++)**

**{**

**c[a[i]-'A']++;**

**d[b[i]-'A']++;**

**}**

**for(int i=0;i<'z'-'A';i++)**

**{**

**if(c[i]!=d[i])**

**return 0;**

**}**

**return 1;**

**}**

**int main(void)**

**{**

**int n=0;**

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

**{**

**char str[n][1000],bro[n][1000],x[1000]={0};**

**int k,key=0;**

**for(int i;i<n;i++)**

**{**

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

**}**

**scanf("%s",&x);**

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

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

**{**

**if(judge\_bianwei(x,str[i]))**

**{**

**strcpy(&bro[key],str[i]);**

**for(int j=key-1;j>=0;j--)**

**{**

**if(strcmp(str[i],bro[j])<0)**

**{**

**strcpy(&bro[j+1],bro[j]);**

**strcpy(&bro[j],str[i]);**

**}**

**}**

**key++;**

**}**

**}**

**printf("%d\n",key);**

**if(k<=key&&k>0)**

**{**

**printf("%s\n",bro[k-1]);**

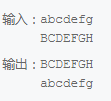
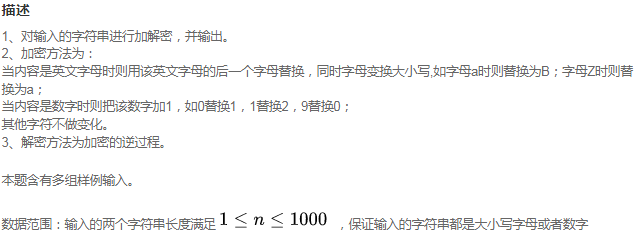
**}**

**}**

**return 0;**

**}**

**字符串加解密**



**输入描述：输入一串要加密的密码**

**输入一串加过密的密码**

**输出描述：输出加密后的字符**

**输出解密后的字符**

**int encode(char \* temp)**

**{**

**char ch;**

**ch = \*temp;**

**if(NULL==temp)**

**{**

**return -1;**

**}**

**if( 'A'<=ch && ch <'Z')**

**{**

**ch = ch +1 +32;**

**}**

**else if('a'<=ch && ch <'z')**

**{**

**ch = ch+1-32;**

**}**

**else if('Z'==ch)**

**{**

**ch='a';**

**}**

**else if('z'==ch)**

**{**

**ch='A';**

**}**

**else if('0'<=ch && ch<'9')**

**{**

**ch = (ch+1);**

**}**

**else if('9'<=ch )**

**{**

**ch = '0';**

**}**

**\*temp = ch;**

**return 0;**

**}**

**int decode(char \* temp)**

**{**

**if(NULL==temp)**

**{**

**return -1;**

**}**

**char ch;**

**ch =\*temp;**

**if( 'A'<ch && ch <='Z')**

**{**

**ch = ch -1 +32;**

**}**

**else if('a'<ch && ch <='z')**

**{**

**ch = ch-1-32;**

**}**

**else if('A'==ch)**

**{**

**ch='z';**

**}**

**else if('a'==ch)**

**{**

**ch='Z';**

**}**

**else if('0'<ch && ch<='9')**

**{**

**ch = ch-1;**

**}**

**else if('0'==ch )**

**{**

**ch = '9';**

**}**

**\*temp=ch;**

**return 0;**

**}**

**int main()**

**{**

**char str1[100];**

**char str2[100];**

**memset(str1,0,sizeof(str1));**

**memset(str2,0,sizeof(str2));**

**while(gets(str1))**

**{**

**gets(str2);**

**for(int i=0;i<strlen(str1);i++)**

**{**

**encode(&str1[i]);**

**}**

**printf("%s\n",str1);**

**for(int i=0;i<strlen(str2);i++)**

**{**

**decode(&str2[i]);**

**}**

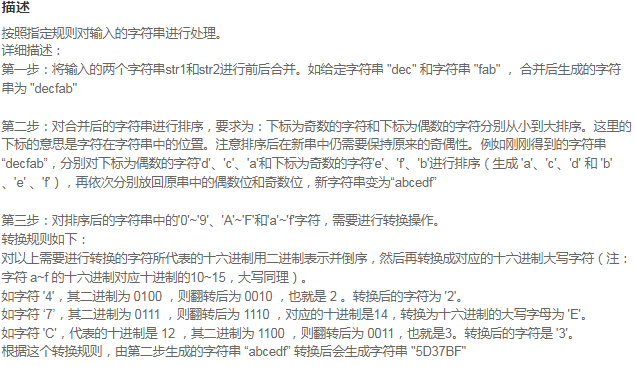
**printf("%s\n",str2);**

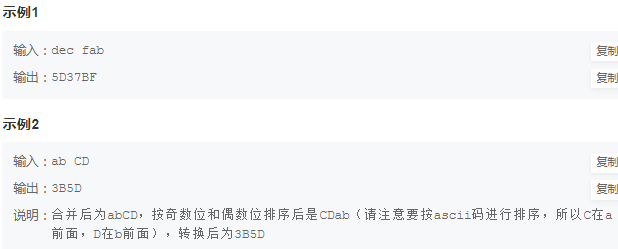
**}**

**return 0;**

**}**

**字符串合并处理：**





**#include <stdio.h>**

**#include <string.h>**

**#include <ctype.h>**

**int dec1(char \*str1)**

**{**

**if(NULL==str1)**

**{**

**return 0;**

**}**

**char temp1[100]={0};**

**char temp2[100]={0};**

**int len1=strlen(str1);**

**int key=0;**

**int point=0;**

**for(int i=0;i<len1;i++)**

**{**

**if(i%2==0)**

**{**

**temp1[key]=str1[i];**

**for(int j=key-1;j>=0;j--)**

**{**

**if(temp1[j]>str1[i])**

**{**

**temp1[j+1]=temp1[j];**

**temp1[j]=str1[i];**

**}**

**}**

**key++;**

**}**

**else if(i%2==1)**

**{**

**temp2[point]=str1[i];**

**for(int j=point-1;j>=0;j--)**

**{**

**if(temp2[j]>str1[i])**

**{**

**temp2[j+1]=temp2[j];**

**temp2[j]=str1[i];**

**}**

**}**

**point++;**

**}**

**}**

**key=0;**

**point=0;**

**for(int i=0;i<len1;i++)**

**{**

**if(i%2==0)**

**{**

**str1[i]=temp1[key];**

**key++;**

**}**

**else if(i%2==1)**

**{**

**str1[i]=temp2[point];**

**point++;**

**}**

**}**

**return 1;**

**}**

**unsigned char str2hex(char ch)**

**{**

**if(isdigit(ch))**

**return ch-'0';**

**else**

**return tolower(ch)-'a'+10;**

**}**

**char hex2str(unsigned char hex)**

**{**

**if(hex<10)**

**return '0'+hex;**

**else**

**return 'A'+hex-10;**

**}**

**unsigned char convertch(char ch)**

**{**

**unsigned char ret=0x00;**

**for(int i=0;i<4;i++)**

**{**

**if(ch&0x1)**

**{**

**ret|=0x1;**

**}**

**ch>>=1;**

**ret<<=1;**

**}**

**ret>>=1;**

**return ret;**

**}**

**void convert(char\* ptr)**

**{**

**int len =strlen(ptr);**

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

**{**

**if(isxdigit(ptr[i]))**

**ptr[i]=hex2str(convertch(str2hex(ptr[i])));**

**}**

**}**

**int main()**

**{**

**char str1[100]={0};**

**char str2[100]={0};**

**char output[200]={0};**

**while(scanf("%s %s",str1,str2)!=EOF)**

**{**

**memset(output,0,sizeof(output));**

**memcpy(output,str1,strlen(str1));**

**strcat(output,str2);**

**if(1!=dec1(output))**

**{**

**return 0;**

**}**

**convert(output);**

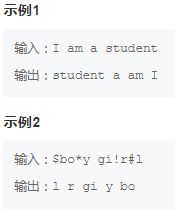
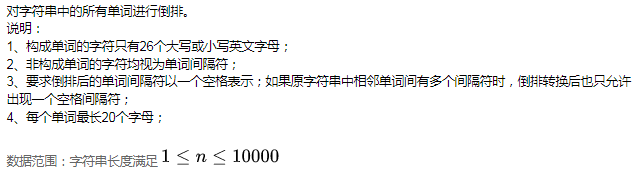
**printf("%s\n",output);**

**}**

**return 0;**

**}**

**单词倒排：**



**int isalp(char ch)**

**{**

**if(('a'<=ch && ch<='z')||('A'<=ch&&ch<='Z'))**

**return 1;**

**else**

**return 0;**

**}**

**int main()**

**{**

**int len=0;**

**int j=0;**

**int num=0;**

**char str[10000]={0};**

**while( gets(str))**

**{**

**len=strlen(str);**

**for(int i=len-1;i>=0;i--)**

**{**

**num=0;**

**if(isalp(str[i]))**

**{**

**for(int j=i;j>=0;j--)**

**{**

**if(str[j]!=' '&& isalp(str[j]))**

**num++;**

**else**

**break;**

**}**

**for(j=i-num+1;j<=i;j++)**

**{**

**printf("%c",str[j]);**

**}**

**printf(" ");**

**}**

**i=i-num;**

**}**

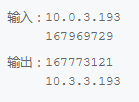
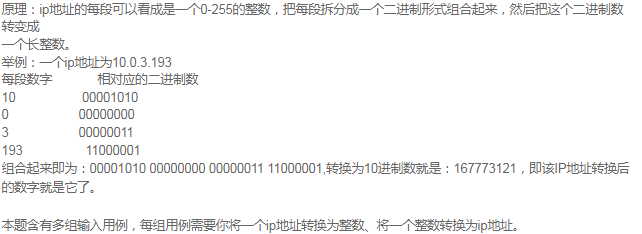
**printf("\n");**

**}**

**return 0;**

**}**

**整数与IP地址间的转换**



**int main()**

**{**

**char str[128];**

**while(gets(str))**

**{**

**int is\_ip=0;**

**int len =strlen(str);**

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

**{**

**if(str[i]=='.')**

**{**

**is\_ip=1;**

**break;**

**}**

**}**

**if(is\_ip)**

**{**

**unsigned int number=0;**

**unsigned char ip[4];**

**sscanf(str,"%d.%d.%d.%d",ip,ip+1,ip+2,ip+3);**

**number|=ip[0]<<24;**

**number|=ip[1]<<16;**

**number|=ip[2]<<8;**

**number|=ip[3];**

**printf("%u\n",number);**

**}**

**else**

**{**

**unsigned int number=0;**

**unsigned char ip[4];**

**sscanf(str,"%u",&number);**

**ip[0]=number>>24&0xff;**

**ip[1]=number>>16&0xff;**

**ip[2]=number>>8&0xff;**

**ip[3]=number&0xff;**

**printf("%d.%d.%d.%d\n",ip[0],ip[1],ip[2],ip[3]);**

**}**

**}**

**return 0;**

**}**