字符数组一本通训练指导

【题目描述】

输入一行字符，统计出其中数字字符的个数。

【输入】

一行字符串，总长度不超过 255。

【输出】

输出为 1 行，输出字符串里面数字字符的个数。

【输入样例】

Peking University is set up at 1898.

【输出样例】

4

5.5-1a

```
#include<iostream>//1. 统计数字字符个数（字符数组）
#include<cstring>// 与 getline 配合使用
using namespace std;
int main()
{
	char ch[256];
	int num=0,l,i;
	cin.getline(ch,256);
	l=strlen(ch); // 计算字符串长度
	for(i=0;i<l;i++)
	{
		if(ch[i]>='0'&&ch[i]<='9')// 判断是否数字字符
			num++;// 累加数字字符个数
	}
	cout<<num;
	return 0;
}
```

5.5-1b

```
#include<iostream>//1. 统计数字字符个数
#include<cstring>
using namespace std;
int main()
{
	string ch;
	int num=0,i;
	cin>>ch;

	for(i=0;i<ch.size();i++)//ch.size() 测量字符串长度
	{
		if(ch[i]>='0'&&ch[i]<='9')// 判断是否数字字符
			num++;// 累加数字字符个数
	}
	cout<<num;
	return 0;
}
```

```
dfg ksas 1235
4
```

【题目描述】

给定一个只包含小写字母的字符串，请你找到第一个仅出现一次的字符。如果没有，输出no。

【输入】一个字符串，长度小于100000。

【输出】输出第一个仅出现一次的字符，若没有则输出no。

【输入样例】abcabd

【输出样例】c

5.5-2a

```
#include<iostream>//2. 第一个只出现一次的字符
#include<cstring>
using namespace std;
char s[100000];
int c[26];
int main()
{
	int i;
	cin>>s;// 读入字符
	for(i=0;i<strlen(s);i++)//strlen(s) 测量字符数组长度
	{
		c[s[i]-'a']++;
	}
	for(i=0;i<strlen(s);i++)
	{
		if(c[s[i]-'a']==1)// 找到第一个仅出现一次的字符
		{
			cout<<s[i];
			return 0;
		}
	}
	cout<<'no';
	return 0;
}
```

5.5-2b//2.c 语言写法

```
#include<cstdio>
#include<cstdlib>// 使用 exit 须调用 cstdlib 库
#include<cstring>
using namespace std;
int main()
{
	int i,j,l,k;
	char a[10000];
	scanf("%s",a);
	l=strlen(a);
	for(i=0;i<l;i++)
	{
		k=0;
		for(j=0;j<l;j++)
		{
			if(a[i]==a[j])  k++;
		}
		if(k==1)
		{
			printf("%c",a[i]);
			exit(0);// 可以用 return 0 代替
		}
	}
	printf("%c","no");
	return 0;
}
```

【题目描述】

为了获知基因序列在功能和结构上的相似性，经常需要将几条不同序列的 DNA 进行比对，以判断该比对的 DNA 是否具有相关性。

现比对两条长度相同的 DNA 序列。定义两条 DNA 序列相同位置的碱基为一个碱基对，如果一个碱基对中的两个碱基相同的话，则称为相同碱基对。接着计算相同碱基对占总碱基对数量的比例，如果该比例大于等于给定阈值时则判定该两条 DNA 序列是相关的，否则不相关。

【输入】

有三行，第一行是用来判定出两条 DNA 序列是否相关的阈值，随后 2 行是两条 DNA 序列（长度不大于 500）。

【输出】

若两条 DNA 序列相关，则输出“yes”，否则输出“no”。

【输入样例】

0.85

ATCGCCGTAAGTAACGGTTTTAAATAGGCC

ATCGCCGGAAGTAACGGTCTTAAATAGGCC

【输出样例】

yes

5.5-3a

```
#include<iostream>//3. 基因相关性
#include<cstring>
using namespace std;
int main()
{
	int i,len,tot=0;//len 字符长度，tot 相似度
	double x;
	char s1[501],s2[502];
	cin>>x>>s1>>s2;
	len=strlen(s1);
	for(i=0;i<len;i++)
	{
		if(s1[i]==s2[i]) tot++;
	}
	if((double)tot/len>=x) cout<<"yes";
	else cout<<"no";
	return 0;
}
```

5.5-3b

```
#include<iostream>//3. 基因相关性
#include<string>
using namespace std;
int main()
{
	string a,b;
	double j,n;// 比例可能为小数，故用 double 类型
	int i,l;
	cin>>n;
	cin>>a>>b;
	l=a.size();// 统计字符数量
	for(i=0;i<l;i++)
	{
		if(a[i]==b[i]) j++;
	}
	if(j/l>=n) cout<<"yes";
	else cout<<"no";
	return 0;
}
```

【题目描述】

游戏规则：石头打剪刀，布包石头，剪刀剪布。现在，需要你写一个程序来判断石头剪子布游戏的结果。

【输入】第一行是一个整数 N，表示一共进行了 N 次游戏。1 ≤ N ≤ 100。

接下来 N 行的每一行包括两个字符串，表示游戏参与者 Player1，Player2 的选择（石头、剪子或者是布）：S1S2 字符串之间以空格隔开 S1,S2 只可能取值在 {"Rock", "Scissors", "Paper"}（大小写敏感）中。

【输出】输出包括 N 行，每一行对应一个胜利者（Player1 或者 Player2），或者游戏出现平局，则输出 Tie。

【输入样例】

```
3
Rock Scissors
Paper Paper
Rock Paper
```

【输出样例】

```
Player1
Tie
Player2
```

```
3
Rock Scissors
Player1
Paper Paper
Tie
Rock Paper
Player2
```

5.5-4a

```
#include<iostream>//4. 剪子包袱锤
#include<string>
using namespace std;
int main()
{
	int i,n;
	cin>>n;
	char a[101],b[101];
	for(i=0;i<n;i++)
	{
		cin>>a>>b;
		if(a[0]=='R'&&b[0]=='S'||a[0]=='S'&&b[0]=='P'||a[0]=='P'&&b[0]=='R')
			cout<<"Player1"<<endl;
		else if(a[0]==b[0]) cout<<"Tie"<<endl;
			else cout<<"Player2"<<endl;
	}//Rock 石头 Scissors 剪刀,Paper 布,tie 不分胜负
	return 0;
}
```

```cpp
//5.5-4b
#include<iostream>
using namespace std;
int main()
{
	int n,i,j,k;
	string a[101],b[101],s[101];
	cin>>n;
	for(i=1;i<=n;i++)
		cin>>a[i]>>b[i];
	for(i=1;i<=n;i++)
	{
		if(a[i]==b[i]) cout<<"Tie"<<endl;
		else if(a[i]=="Rock"&&b[i]=="Scissors"
			||a[i]=="Scissors"&&b[i]=="Paper"
			||a[i]=="Paper"&&b[i]=="Rock")  cout<<"Player1"<<endl;
			else cout<<"Player2"<<endl;
	}
	return 0;
}
```

```
3
Rock Scissors
Paper Paper
Rock Paper
Player1
Tie
Player2
```

【题目描述】

编写程序，求给定字符串 s 的亲朋字符串 s1。

亲朋字符串 s1 定义如下：给定字符串 s 的第一个字符的 ASCII 值加第二个字符的 ASCII 值，得到第一个亲朋字符； 给定字符串 s 的第二个字符的 ASCII 值加第三个字符的 ASCII 值，得到第二个亲朋字符；依此类推，直到给定字符串 s 的倒数第二个字符。亲朋字符串的最后一个字符由给定字符串 s 的最后一个字符 ASCII 值加 s 的第一个字符的 ASCII 值。

【输入】

输入一行，一个长度大于等于 2，小于等于 100 的字符串。字符串中每个字符的 ASCII 值不大于 63。

【输出】

输出一行，为变换后的亲朋字符串。输入保证变换后的字符串只有一行。

【输入样例】

1234

【输出样例】

cege

5.5-5

```
#include<iostream>//5. 输出亲朋字符串
#include<cstring>
using namespace std;
char a[100];
int main()
{
	int i,l;
	cin>>a;
	l=strlen(a);
	for(i=0;i<=l-2;i++)
	{
		cout<<a[i]+a[i+1];
	}
	cout<<a[l-1]+a[0];
	return 0;
}
```

【题目描述】

给定一个不包含空白符的字符串，请判断是否是 C 语言合法的标识符号（注：题目保证这些字符串一定不是 C 语言的保留字）。

C 语言标识符要求：1. 非保留字；2. 只包含字母、数字及下划线（“_”）。3. 不以数字开头。

【输入】一行，包含一个字符串，字符串中不包含任何空白字符，且长度不大于 20。

【输出】一行，如果它是 C 语言的合法标识符，则输出 yes，否则输出 no。

【输入样例】RKPEGX9R;TWyYcp　　【输出样例】no

5.5-6

```
#include<iostream>//6. 合法 c 标识符
#include<cstring>
using namespace std;
int main()
{
	int i,l,h;
	char a[256];
	bool f=true,b=true;
	cin>>a;
	l=strlen(a)-1;
	if(a[0]>='0'&&a[0]<='9') b=false;// 如果数字开头，不是合法标识符
	for(i=0;i<=l;i++)
	{
		if(a[0]>='0'&&a[i]<='9'||a[i]>='a'&&a[i]<='z'
		||a[i]>='A'&&a[i]<='Z'||a[i]>='_');
		// 如果只包含字母、数字及下划线，为合法标识符
		else f=false;
	}
	if(f==true&&b==true) cout<<"yes";
	else cout<<"no";;
	return 0;
}
```

5.5-6b

```
#include<iostream>//6. 合法 c 标识符
#include<cstring>
using namespace std;
int main()
{
	int i,l,h;
	char a[256];
	cin>>a;
	if(a[0]>='0'&&a[0]<='9')
	{
		cout<<"no";
		return 0;
	}
	l=strlen(a);
	for(i=0;i<l;i++)
	{
		if(!(a[i]>='0'&&a[i]<='9'||a[i]>='A'&&a[i]<='Z'
		||a[i]>='a'&&a[i]<='z'))
		{
			cout<<"no";
			return 0;
		}
	}
	cout<<"yes";
	return 0;
}
```

【题目描述】

脱氧核糖核酸（DNA）由两条互补的碱基链以双螺旋的方式结合而成。而构成 DNA 的碱基共有 4 种，分别为腺嘌呤（A）、鸟嘌呤（G）、胸腺嘧啶（T）和胞嘧啶（C）。我们知道，在两条互补碱基链的对应位置上，腺嘌呤总是和胸腺嘧啶配对，鸟嘌呤总是和胞嘧啶配对。你的任务就是根据一条单链上的碱基序列，给出对应的互补链上的碱基序列。

【输入】

一个字符串，表示一条碱基链。这个字符串只含有大写字母 A、T、G、C，分别表示腺嘌呤、胸腺嘧啶、鸟嘌呤和胞嘧啶。字符串长度不超过 255。

【输出】

一个只含有大写字母 A、T、G、C 的字符串，为与输入的碱基链互补的碱基链。

【输入样例】

ATATGGATGGTGTTTGGCTCTG

【输出样例】

TATACCTACCACAAACCGAGAC

5.5-7a

```
#include<iostream>//7. 配对碱基链
#include<string>
using namespace std;
int main()
{
	string s;
	int i,l;
	cin>>s;

	for(i=0;i<s.size();i++)
	{
		if(s[i]=='A') cout<<"T";
		if(s[i]=='T') cout<<"A";
		if(s[i]=='G') cout<<"C";
		if(s[i]=='C') cout<<"G";
	}
	return 0;
}
```

5.5-7b

```
#include<iostream>//7. 配对碱基链
#include<cstring>
using namespace std;
int main()
{
	char s[256];
	int i;
	cin>>s;

	for(i=0;i<strlen(s);i++)//strlen(s) 测量字符数组长度
	{
		if(s[i]=='A') cout<<"T";
		if(s[i]=='T') cout<<"A";
		if(s[i]=='G') cout<<"C";
		if(s[i]=='C') cout<<"G";
	}
	return 0;
}
```

【题目描述】

在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。我们给出一种最简的的加密方法，对给定的一个字符串，把其中从 a-y，A-Y 的字母用其后继字母替代，把 z 和 Z 用 a 和 A 替代，其他非字母字符不变，则可得到一个简单的加密字符串。

【输入】输入一行，包含一个字符串，长度小于 80 个字符。

【输出】输出每行字符串的加密字符串。

【输入样例】Hello! How are you!

【输出样例】Ifmmp! Ipx bsf zpv!

5.5-8a

```
#include<iostream>//8. 密码翻译
#include<cstring>
using namespace std;
char s[80];
int i,l;
int main()
{
	cin.getline(s,80);
	l=strlen(s);
	for(i=0;i<l;i++)
	{
		if(s[i]=='z'||s[i]=='Z')
			s[i]=s[i]-25;
		else if(s[i]>='a'&&s[i]<='z'||s[i]>='A'&&s[i]<='Z')
			s[i]=s[i]+1;
		cout<<s[i];
	}
	return 0;
}
```

5.5-8b

```
#include<iostream>//8. 密码翻译
#include<string>
using namespace std;
int main()
{
	int i,len;
	string s;
	getline(cin,s);// 可以输入空格
	len=s.size();
	for(i=0;i<len;i++)
	{
		if(s[i]=='z'||s[i]=='Z')
			s[i]=s[i]-25;
		else if(s[i]>='a'&&s[i]<'z'||s[i]>='A'&&s[i]<'Z')
			s[i]=s[i]+1;
		cout<<s[i];
	}
	return 0;
}
```

【题目描述】

小英是药学专业大三的学生，暑假期间获得了去医院药房实习的机会。

在药房实习期间，小英扎实的专业基础获得了医生的一致好评，得知小英在计算概论中取得过好成绩后，主任又额外交给她一项任务，解密抗战时期被加密过的一些伤员的名单。

经过研究，小英发现了如下加密规律（括号中是一个“原文 -> 密文”的例子）

1. 原文中所有的字符都在字母表中被循环左移了三个位置（dec -> abz）2. 逆序存储（abcd -> dcba ）3. 大小写反转（abXY -> ABxy）

【输入】一个加密的字符串。（长度小于 50 且只包含大小写字母）

【输出】输出解密后的字符串。

【输入样例】GSOOWFASOq

【输出样例】Trvdizrrvj

5.5-9b

```
#include<iostream>//9.b 加密的病历单
#include<cstring>
using namespace std;
int main()
{
	char a[50];
	int i,l;
	cin>>a;
	l=strlen(a);
	for(i=0;i<l;i++)
	{
		if(a[i]>='x'&&a[i]<='z'||a[i]>='X'&&a[i]<='Z')
			a[i]=a[i]-23;
		else a[i]=a[i]+3;
		if(a[i]>='A'&&a[i]<='Z')
			a[i]=a[i]+32;// 大小写转化
		else a[i]=a[i]-32;
	}
	for(i=l-1;i>=0;i--) cout<<a[i];
	return 0;
}
```

5.5-9a

```
#include<iostream>//9. 加密的病历单
#include<cstring>
using namespace std;
int main()
{
	int i,len,t;
	char s[10001];
	cin>>s;
	len=strlen(s);
	for(i=0;i<len;i++)
	{
		if(s[i]>='a'&&s[i]<='z')
		{
			t=s[i];
			t+=3;// 向右移动 3 个位置恢复原文
			if(t>122) t-=26;// 如果是 x,y,z, 就恢复为 a,b,c
			t-=32;s[i]=t;// 小写转为大写
		}
		else if(s[i]>='A'&&s[i]<='Z')// 如果是大写
		{
			t=s[i];t+=3;// 向右移动三个位置恢复原文
			if(t>90) t-=26;// 若为 X,Y,Z, 恢复为 A,B,C
			t+=32;s[i]=t;// 大写转为小写
		}
	}
	for(i=len-1;i>=0;i--) cout<<s[i];// 逆序输出
	return 0;
}
```

【题目描述】

给定一个字符串，将其中所有的小写字母转换成大写字母。

【输入】

输入一行，包含一个字符串（长度不超过 100，可能包含空格）。

【输出】

输出转换后的字符串。

【输入样例】

helloworld123Ha

【输出样例】

HELLOWORLD123HA

5.5-10a

```
#include<iostream>//10.a 小写转大写
#include<string>
using namespace std;
string s;
int i,len;
int main()
{
	getline(cin,s);//getline 可以输入空格
	len=s.size();// 获取字符串长度
	for(i=0;i<len;i++)
	{
		if(s[i]>='a'&&s[i]<='z')
			s[i]=s[i]-32;// 如果为小写，转为大写
	}
	cout<<s;
	return 0;
}
```

5.5-10b

```
#include<iostream>//10.b 小写转大写 仅做参考
#include<cstring>
using namespace std;
char s[100];
int i,len;
int main()
{
	cin>>s;
	len=strlen(s);
	for(i=0;i<len;i++)
	{
		if(islower(s[i]))
			s[i]=toupper(s[i]);
	}
	cout<<s;
	return 0;
}
```

【题目描述】医生在书写药品名的时候经常不注意大小写，格式比较混乱。现要求你写一个程序将医生书写混乱的药品名整理成统一规范的格式，即药品名的第一个字符如果是字母要大写，其他字母小写。如将 ASPIRIN、aspirin 整理成 Aspirin。

【输入】第一行一个数字 n，表示有 n 个药品名要整理，n 不超过 100。

接下来 n 行，每行一个单词，长度不超过 20，表示医生手书的药品名。药品名由字母、数字和 - 组成。

【输出】n 行，每行一个单词，对应输入的药品名的规范写法。

5.5-11a

```
#include<iostream>//11.a 整理药名 仅做参考
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
	char str[25];
	int n,len;
	cin>>n;
	for(int i=0; i<n; i++)
	{
		scanf("%s",str);
		if(str[0]>='a'&&str[0]<='z')// 以小写开头，转为大写
			str[0]-=32;
		len=strlen(str);
		for(int j=1; j<len; j++)
		{ // 除开头外均转为小写字母
			if(str[j]>='A'&&str[j]<='Z')
			{
				str[j]+=32;
			}
		}
		puts(str);
	}
	return 0;
}
```

【输入样例】

```
4
AspiRin
cisapride
2-PENICILLIN
Cefradine-6
```

【输出样例】

```
Aspirin
Cisapride
2-penicillin
Cefradine-6
```

5.5-11b

```
#include<iostream>//11.b 整理药名
#include<string>
using namespace std;
int main()
{
    string s;
    int n;
    cin>>n;
    while(n--)
    {
      cin>>s;
      if(s[0]>='a'&&s[0]<='z')
            s[0]-=32;
      for(int i=1;i<s.size();i++)
      {
            if(s[i]>='A'&&s[i]<='Z')
                  s[i]+=32;
            }
            cout<<s<<endl;
      }
      return 0;
}
```

5.5-11c

```
#include<iostream>//11.c 整理药名
#include<string>
using namespace std;
int main()
{
    string s;
    int n;
    cin>>n;
    for(int j=1;j<=n;j++)
    {
      cin>>s;
      if(s[0]>='a'&&s[0]<='z')
            s[0]-=32;
      for(int i=1;i<s.size();i++)
      {
            if(s[i]>='A'&&s[i]<='Z')
                  s[i]+=32;
            }
            cout<<s<<endl;
      }
      return 0;
}
```

【题目描述】

输入两个字符串，验证其中一个串是否为另一个串的子串。

【输入】

输入两个字符串， 每个字符串占一行，长度不超过 200 且不含空格。

【输出】

若第一个串 s1 是第二个串 s2 的子串，则输出 (s1) is substring of (s2)

否则，若第二个串 s2 是第一个串 s1 的子串，输出 (s2) is substring of (s1)

否则，输出 No substring。

【输入样例】abc dddncabca

【输出样例】abc is substring of dddncabca

5.5-12a

```
#include<cstdio>//12. 验证子串
#include<cstdlib>
#include<cstring>
using namespace std;
char s1[200],s2[200];
int main()
{
	int i,n,t;
	scanf("%s",s1);
	scanf("%s",s2);
	if(strstr(s2,s1))//strstr 函数判断子串
	{
		printf("%s is substring of %s",s1,s2);
		exit(0);// 退出
	}
	if(strstr(s1,s2))
	{
		printf("%s is substring of %s",s2,s1);
		exit(0);
	}
	printf("No substring");
	return 0;
}
```

5.5-12b

```
#include<iostream>//12b, 验证子串
using namespace std;
string a,b;
int main()
{
	cin>>a>>b;
	if(a.find(b)!=a.npos)// 使用函数查找
		cout<<b<<" is substring of "<<a;
	else if(b.find(a)!=b.npos)
		cout<<a<<" is substring of "<<b;
	else cout<<"No substring";
	return 0;
}
```

```
abc
dsfgdfgdrfgabcrger
abc is substring of dsfgdfgdrfgabcrger
```

```
abc
sdfsfsdfds
No substring
```

【题目描述】

给定一个单词，如果该单词以 er、ly 或者 ing 后缀结尾， 则删除该后缀（题目保证删除后缀后的单词长度不为 0）， 否则不进行任何操作。

【输入】输入一行，包含一个单词（单词中间没有空格，每个单词最大长度为 32）。

【输出】输出按照题目要求处理后的单词。

【输入样例】referer

【输出样例】refer

5.5-13a1

```
#include<iostream>//13a2 删除单词后缀 来源于 csdn
#include<string>
using namespace std;
int main()
{
    string s;
    int len;

    cin>>s;
    len=s.size();
    if((s[len-2]=='e'&& s[len-1]=='r')||(s[len-2]=='l'&&s[len-1]=='y'))
    {
      for(int i=0;i<len-2;i++)
            cout<<s[i];
      }
      else if(s[len-3]=='i'&&s[len-2]=='n'&&s[len-1]=='g')
      {
            for(int i=0;i<len-3;i++)
                  cout<<s[i];
      }
      else
            cout<<s<<endl;
      return 0;
}
```

5.5-13a2

```
#include<iostream>
#include<cstdio>//13a. 删除单词后缀
#include<cstring>
using namespace std;
int i,len;
char s[32];
int main()
{
	scanf("%s",s);
	len=strlen(s);
	if(strcmp(&s[len-3],"ing")==0) s[1-3]='\0';
	//strcmp 函数判定字符数组后三位是否为 ing, 若是删除
	if((strcmp(&s[len-2],"er")==0)||(strcmp(&s[len-2],"ly")==0))
s[len-2]='\0';
	printf("%s",s);
	return 0;
}
```

5.5-13b

```
#include<iostream>//13b. 删除单词后缀
#include<cstring>
using namespace std;
int i,len;
string s;
int main()
{
	getline(cin,s);// 这个输入方法比较稳妥，可以输入空格
	len=s.size();
	if(strcmp(&s[len-3],"ing")==0) len-=3;
	else if((strcmp(&s[len-2],"er")==0)||(strcmp(&s[len-2],"ly")==0)) len-=2;
	for(i=0;i<len;i++) cout<<s[i];
	return 0;
}
```

【题目描述】

输入一行单词序列，相邻单词之间由 1 个或多个空格间隔，请对应地计算各个单词的长度。注意：如果有标点符号（如连字符，逗号），标点符号算作与之相连的词的一部分。没有被空格间开的符号串，都算作单词。

【输入】

一行单词序列，最少 1 个单词，最多 300 个单词，单词之间用至少 1 个空格间隔。单词序列总长度不超过 1000。

【输出】

依次输出对应单词的长度，之间以逗号间隔。

【输入样例】

She was born in 1990-01-02 and from Beijing city.

【输出样例】

3,3,4,2,10,3,4,7,5

5.5-14b

```
#include<iostream>//14b 单词长度
using namespace std;
int i,n,t;
bool f=1;
string s;
int main()
{
	while(cin>>s)
	{
		if(f) f=0;
		else cout<<" ";
		cout<<s.size();
	}
	return 0;
}
```

1143：最长最短单词

时间限制： 1000 ms　　内存限制： 65536 KB

提交数： 26726　　通过数： 9724

【题目描述】

输入 1 行句子（不多于 200 个单词，每个单词长度不超过 100），只包含字母、空格和逗号。单词由至少一个连续的字母构成，空格和逗号都是单词间的间隔。

试输出第 1 个最长的单词和第 1 个最短单词。

【输入】

一行句子。

【输出】

第 1 行，第一个最长的单词。

第 2 行，第一个最短的单词。

【输入样例】

I am studying Programming language C in Peking University

【输出样例】

Programming

I

【提示】

如果所有单词长度相同，那么第一个单词既是最长单词也是最短单词。

5.5-15a

```
#include<iostream>//15a. 最长最短单词
#include<cstring>
using namespace std;
int main()
{
	char s[25000];
	int i,l,t=0,maxl=0,minl=101,maxi,mini;
	cin.getline(s,25000);
	l=strlen(s);
	s[l]=' ';// 最后添加空格，方便判定单词
	for(i=0;i<=l;i++)
	{
		if(s[i]!=' '&&s[i]!=',') t++;
		else if(t>0)
		{
			if(t>maxl)
			{
				maxl=t;
				maxi=i-t;
			}
			if(t<minl)
			{
				minl=t;
				mini=i-t;
			}
			t=0;
		}
	}
	for(i=maxi;i<=maxi+maxl-1;i++)
	{cout<<s[i];}
	cout<<endl;
	for(i=mini;i<=mini+minl-1;i++){cout<<s[i];}
	return 0;
}
```

5.5-15b

```
#include<iostream>//15b 最长最短单词
using namespace std;
int main()
{
	string s;
	int i,t=0,l,maxl=0,minl=101,maxi,mini;
	getline(cin,s);
	l=s.size();
	s[l]=' ';// 字符串后加一个空格，方便判定最后一个单词
	for(i=0;i<=l;i++)
	{
		if(s[i]!=' '&&s[i]!=',') t++;// 累加单词长度
		else if(t>0)
		{
			if(t>maxl)
			{
				maxl=t;
				maxi=i-t;// 最长单词起始位置
			}
			if(t<minl)
			{
				minl=t;
				mini=i-t;// 最小单词起始位置
			}
			t=0;
		}
	}
	for(i=maxi;i<=maxi+maxl-1;i++)
	{cout<<s[i];}
	cout<<endl;
	for(i=mini;i<=mini+minl-1;i++)
	{cout<<s[i];}
	return 0;
}
```

1144：单词翻转

时间限制： 1000 ms　　内存限制： 65536 KB

提交数： 18986　　通过数： 9390

【题目描述】输入一个句子（一行），将句子中的每一个单词翻转后输出。

【输入】只有一行，为一个字符串，不超过 500 个字符。单词之间以空格隔开。

【输出】翻转每一个单词后的字符串，单词之间的空格需与原文一致。

【输入样例】hello world

【输出样例】olleh dlrow

5.5-16b

```
#include<iostream>//16b. 单词翻转
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
	char s[501];
	int i,l,j,x=0,y;
	cin.getline(s,501);
	l=strlen(s);
	s[l]=' ';// 在末尾增加一个空格方便计算
	for(i=0;i<=l;i++)
	{
		if(s[i]!=' ') x=x+1;// 计算单词长度
		else
		{
			y=i;// 单词末尾的位置
			for(j=1;j<=x;j++) cout<<s[--y];// 从后往前输出
			x=0;
			if(i!=l) cout<<" ";// 如果不是自己加上的空格就输出
		}
	}
	return 0;
}
```

5.5-16a

```
#include<iostream>
#include<cstdio>//16a. 单词反转
#include<cstring>
using namespace std;
char s[1001];
int main()
{
	cin.getline(s,1001);
	int i,l,j;
	l=strlen(s);
	for(i=0;i<l;i++)
	{
		if(s[i]==' ') printf(" ");// 保留所有空格
		else
		{
			int k=0;// 单词长度清零
			for(j=i;s[j]!=' ';j++)
			{
				k++;
				if(j>=l) break;
			}
			j--;
			k--;
			for(j=j;j>=i;j--) printf("%c",s[j]);// 单词倒序输出
			i+=k;
		}
	}
	return 0;
}
```

【题目描述】

给定一个完全由数字字符（‘0’，‘1’，‘2’，…，‘9’）构成的字符串 str，请写出 str 的 p 型编码串。例如：字符串 122344111 可被描述为"1 个 1、2 个 2、1 个 3、2 个 4、3 个 1"，因此我们说 122344111 的 p 型编码串为 1122132431；类似的道理，编码串 101 可以用来描述 1111111111；00000000000 可描述为"11 个 0"，因此它的 p 型编码串即为 110；100200300 可描述为"1 个 1、2 个 0、1 个 2、2 个 0、1 个 3、2 个 0"，因此它的 p 型编码串为 112012201320。

【输入】输入仅一行，包含字符串 str。每一行字符串最多包含 1000 个数字字符。

【输出】输出该字符串对应的 p 型编码串。

【输入样例】122344111

【输出样例】1122132431

5.5-17a

```
#include<iostream>//17a 字符串 p 型编码
#include<cstring>
using namespace std;
int main()
{
	char s[1001];
	int i,t=1,l;
	cin>>s;
	l=strlen(s);
	for(i=0;i<l;i++)
	{
		if(s[i]==s[i+1]) t++;
		else
		{
			cout<<t<<s[i];
			t=1;
		}
	}
	return 0;
}
```

5.5-17b

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
	string s;
	int i,t=1,l;
	cin>>s;
	l=s.size();
	for(i=0;i<l;i++)
	{
		if(s[i]==s[i+1]) t++;
		else
		{
			cout<<t<<s[i];
			t=1;
		}
	}
	return 0;
}
```

1146：判断字符串是否为回文

时间限制： 1000 ms　　内存限制： 65536 KB

提交数： 18324　　通过数： 11354

【题目描述】

输入一个字符串，输出该字符串是否回文。回文是指顺读和倒读都一样的字符串。

【输入】输入为一行字符串（字符串中没有空白字符，字符串长度不超过 100）。

【输出】如果字符串是回文，输出 yes；否则，输出 no。

【输入样例】abcdedcba

【输出样例】yes

5.5-18a

```
#include<iostream>//18a 字符串回文
#include<cstring>
using namespace std;
char s[100];
int main()
{
	cin.getline(s,100);
	int p=strlen(s)-1;
	int j=0;//p 为尾，j 为首，从字符串首尾同时往中间判断
	while(j<p&&s[j]==s[p])
	{
		p--;
		j++;
	}
	if(j>=p) cout<<"yes";
	else cout<<"no";
	return 0;
}
```

5.5-18b

```
#include<iostream>//18b. 字符串回文
using namespace std;
string s;
int main()
{
	getline(cin,s);
	int p=s.size()-1;
	int j=0;
	while(j<p&&s[j]==s[p])
	{
		p--;
		j++;
	}
	if(j>=p) cout<<"yes";
	else cout<<"no";
	return 0;
}
```

1147：最高分数的学生姓名

时间限制： 1000 ms　　内存限制： 65536 KB　　提交数： 14826　　通过数： 10512

【题目描述】输入学生的人数，然后再输入每位学生的分数和姓名，求最高分数的学生的姓名。

【输入】第一行输入一个正整数N（N <= 100），表示学生人数。接着输入N行，每行格式如下：

分数 姓名

分数是一个非负整数，且小于等于100；姓名为一个连续的字符串，中间没有空格，长度不超过20。数据保证最高分只有一位同学。

【输出】获得最高分数同学的姓名。

【输入样例】

```
5
87 lilei
99 hanmeimei
97 lily
96 lucy
77 jim
```

【输出样例】

```
hanmeimei
```

5.5-19a

```
#include<iostream>//19a 最高分数学生
using namespace std;
int main()
{
	int a,maxn=-1,i,n,t;
	string s,maxs;
	cin>>n;
	for(i=1;i<=n;i++)
	{
		cin>>a>>s;
		if(a>maxn)
		{
			maxn=a;
			maxs=s;
		}
	}
	cout<<maxs;
	return 0;
}
```

1148：连续出现的字符

时间限制： 1000 ms　　内存限制： 65536 KB

提交数： 23024　　通过数： 9016

【题目描述】给定一个字符串，在字符串中找到第一个连续出现至少 k 次的字符。

【输入】第一行包含一个正整数 k，表示至少需要连续出现的次数。1 ≤ k ≤ 1000。第二行包含需要查找的字符串。字符串长度在 1 到 2500 之间，且不包含任何空白符。

【输出】若存在连续出现至少 k 次的字符，输出该字符；否则输出 No。

【输入样例】

3

abcccaaab

【输出样例】

c

5.5-20a

```
#include<iostream>//20a 连续出现的字符　仅做参考
#include<cstring>
using namespace std;
int main()
{
	int i,j,k,t;
	char s[1000];
	cin>>k;
	cin>>s;
	int l=strlen(s);
	for(i=0;i<=l;i++)
	{
		t=i;
		while(s[i]==s[t]) t++;
		if(t-i==k)
		{
			cout<<s[i];return 0;// 找到后输出，立即退出
		}
	}
	cout<<"no";
	return 0;
}
```

5.5-20b

```
#include<iostream>//20b 连续出现的字符
#include<cstring>
using namespace std;
int main()
{
	int i,j,k,t,l;
	char s[1000];
	cin>>k;
	cin>>s;
	l=strlen(s);
	for(i=0;i<l;i++)
	{
		if(t==k)
		{
			cout<<s[i];return 0;
		}
		if(s[i]==s[i+1]) t++;
		else t=1;
	}
	cout<<"no";
	return 0;
}
```

1149：最长单词 2

时间限制： 1000 ms　　内存限制： 65536 KB

【题目描述】一个以‘.’结尾的简单英文句子，单词之间用空格分隔，没有缩写形式和其它特殊形式。

【输入】一个以‘.’结尾的简单英文句子（长度不超过 500），单词之间用空格分隔，没有缩写形式和其它特殊形式。

【输出】该句子中最长的单词。如果多于一个，则输出第一个。

【输入样例】I am a student of Peking University.

【输出样例】University

5.5-21b

```
#include<iostream>//21b 最长单词
#include<cstring>
using namespace std;
char s[20001];
int main()
{
	int t=0,l,maxl=0,maxi,i;
	cin.getline(s,20001);
	l=strlen(s);
	for(i=0;i<l;i++)
	{
		if(s[i]!=' '&&s[i]!='.') t++;// 如果不是空格或者句号，就累加长度
		else if(t>0)
		{
			if(t>maxl)// 如果当前单词长度大于最长单词
			{
				maxl=t;
				maxi=i-t;// 记下最长单词的起始位置
			}
			t=0;
		}
	}
	for(i=maxi;i<=maxi+maxl-1;i++) cout<<s[i];
	return 0;
}
```

5.5-21c

```
#include<bits/stdc++.h>//21b 最长单词 csdn
using namespace std;
int main()
{
    char a[600];
    gets(a);//gets 已经取消，仅做解释
    int max=0,max1,max2; //1 记录起始位置，2 记录结束位置
    int sum=0;
    int n=strlen(a); // 计算长度
    for(int i=0;i<n;i++)
    {
        if(a[i]!=' '&& a[i]!='.') // 如果不是空格和句号
            sum++; // 计数器加一
        else // 否则
        {
            if(sum>max) // 如果大于最大
            {
                max=sum; // 赋值
                max1=i-sum; // 计算位置
                max2=i-1;
            }
            sum=0; // 归 0
        }
    }
    for(int i=max1;i<=max2;i++)
        cout<<a[i]; // 输出单词
    return 0; // 养成好习惯
}
```

字符串一本通训练指导

读入未知数目的 string 对象，每次输出对应的 string 对象和编号

5.6-1a

```
#include<iostream>//1a. 输出对应 string 编号和内容
using namespace std;
int main()
{
	string str;
	int tot=0;
	while(cin>>str)
	{
		cout<<tot++<<" "<<str<<endl;
	}
	return 0;
}
```

```
welcome to my world
0 welcome
1 to
2 my
3 world
```

5.6-1b

```
#include<iostream>//++ 在前
using namespace std;
int main()
{
	string str;
	int tot=0;
	while(cin>>str)
	{
		cout<<++tot<<" "<<str<<endl;
	}
	return 0;
}
```

```
welcome to my world
1 welcome
2 to
3 my
4 world
```

2047：【例 5.16】过滤空格

时间限制： 1000 ms　　内存限制： 65536 KB

【题目描述】

过滤多余的空格。一个句子中也许有多个连续空格，过滤掉多余的空格，只留下一个空格。

【输入】

一行，一个字符串（长度不超过 200200），句子的头和尾都没有空格。

【输出】

过滤之后的句子。

【输入样例】

```
Hello      world.This is     c language.
```

【输出样例】

```
Hello world.This is c language.
```

5.6-2a

```
#include <iostream>//2. 过滤所有的空格 仅做参考
using namespace std;
int main()
{
      string a;
    getline(cin,a);
    for (int i=0; i<a.size(); i++)
      {
            if(a[i]!=' ')// 如果不等于空格就输出
            {
                  cout<<a[i];
            }
    }
    return 0;
}
```

```
Hello      world.This is
c language.


Helloworld.Thisisclanguage.
```

5.6-2b

```
#include<iostream>//2. 过滤多余的空格
using namespace std;
int main()
{
      string a;
    getline(cin,a);
    for(int i=0; i<a.size(); i++)
      {
          // 对每一个字符做判断，如果当前字符的后一个字符不等于空格，或者自己不等于空格就输出
            // 自己是空格，后面是空格，不能输出
            if(a[i+1]!=' '||a[i]!=' ')
            {
                  cout<<a[i];
            }
    }
    return 0;
}
```

```
Hello      world.This is    c language.

Hello world.This is c language.
```

描述

把一个字符串中所有出现的大写字母都替换成小写字母，同时把小写字母替换成大写字母。

输入

输入一行：待互换的字符串。

输出

输出一行：完成互换的字符串（字符串长度小于 80）。

样例输入

```
If so, you already have a Google Account. You can sign in on the right.
```

样例输出

```
iF SO, YOU ALREADY HAVE A gOOGLE aCCOUNT. yOU CAN SIGN IN ON THE RIGHT.
```

5.6-3

```
#include<iostream>// 大小写互换
using namespace std;
string s;
int main()
{
	getline(cin,s);
	for(int i=0;i<s.size();i++)
	{
		if(s[i]>='A'&&s[i]<='Z') s[i]+=('a'-'A');
		else if(s[i]>='a'&&s[i]<='z') s[i]-=('a'-'A');// 大小写互换
	}
	cout<<s;
	return 0;
}
```

一般我们用 strcmp 可比较两个字符串的大小，比较方法为对两个字符串从前往后逐个字符相比较（按 ASCII 码值大小比较），直到出现不同的字符或遇到’\0’为止。如果全部字符都相同，则认为相同；如果出现不相同的字符，则以第一个不相同的字符的比较结果为准（注意：如果某个字符串遇到’\0’，而另一个字符串还未遇到’\0’，则前者小于后者）。但在有些时候，我们比较字符串的大小时，希望忽略字母的大小，例如 "Hello" 和 "hello" 在忽略字母大小写时是相等的。请写一个程序，实现对两个字符串进行忽略字母大小写的大小比较。

输入格式　输入为两行，每行一个字符串，共两个字符串。（每个字符串长度都小于 80 且只包含大小写字母）

输出格式　如果第一个字符串比第二个字符串小，输出一个字符 "<"；　如果第一个字符串比第二个字符串大，输出一个字符 ">"；

如果两个字符串相等，输出一个字符 "="。

样例输入	样例输出
Hellohowareyou helloHowareyou	=

5.6-4

```
#include<iostream>//4. 忽略大小写的字符串比较
using namespace std;
string s1,s2;
int main()
{
	getline(cin,s1);
	getline(cin,s2);
	for(int i=0;i<s1.size();i++)
	{
		if(s1[i]>='a'&&s1[i]<='z') s1[i]-=('a'-'A');// 小写转大写
	}
	for(int i=0;i<s2.size();i++)
	{
		if(s2[i]>='a'&&s2[i]<='z') s2[i]-=('a'-'A');
	}
	if(s1>s2) cout<<">";
	else if(s1<s2) cout<<"<";
		else cout<<"=";
}
```

【题目描述】

判断两个由大小写字母和空格组成的字符串在忽略大小写，且忽略空格后是否相等。

【输入】两行，每行包含一个字符串。

【输出】若两个字符串相等，输出 YES，否则输出 NO。

【输入样例】

a A bb BB ccc CCC

Aa BBbb CCCccc

【输出样例】

YES

5.6-5

```
#include<iostream>//5. 字符串判等
using namespace std;
string x,y,p,q;
int main()
{
	getline(cin,x);
	getline(cin,y);
	p=q="";// 初始化 p,q 为空串
	for(int i=0;i<x.size();i++)
	{
		if(x[i]>='A'&&x[i]<='Z') x[i]+='a'-'A';// 大写转小写
		if(x[i]!=' ') p+=x[i];
	}
	for(int i=0;i<y.size();i++)
	{
		if(y[i]>='A'&&y[i]<='Z') y[i]+='a'-'A';
		if(y[i]!=' ') q+=y[i];
	}
	if(p==q) cout<<"yes";
	else cout<<"no";
	return 0;
}
```

【题目描述】字符串移位包含问题。对于一个字符串来说，定义一次循环移位操作为：将字符串的第一个字符移动到末尾形成新的字符串。

给定两个字符串 s1s1 和 s2s2，要求判定其中一个字符串是否是另一字符串通过若干次循环移位后的新字符串的子串。例如 CDAA 是由 AABCD 两次移位后产生的新串 BCDAA 的子串，而 ABCD 与 ACBD 则不能通过多次移位来得到其中一个字符串是新串的子串。

【输入】一行，包含两个字符串，中间由单个空格隔开。字符串只包含字母和数字，长度不超过 3030。

【输出】如果一个字符串是另一字符串通过若干次循环移位产生的新串的子串，则输出 true，否则输出 false。

【输入样例】AABCD CDAA　　【输出样例】true

5.6-6

```
#include<iostream>//6. 字符串移位包含问题
using namespace std;
string x,y;
char z;
int main()
{
	cin>>x>>y;
	if(x.length()<y.length()) swap(x,y);
	int l=x.length();
	for(int i=0;i<l;i++)
	{
		z=x[0];
		x=x.substr(1,l);// 等同于 x.erase(0,1)
		x+=z;
		if(x.find(y,0)!=string::npos)
		{
			cout<<"true";
			return 0;
		}
	}
	cout<<"false";
	return 0;
}
```

【题目描述】输入一个字符串，以回车结束（字符串长度≤ 200）。该字符串由若干个单词组成，单词之间用一个空格隔开，所有单词区分大小写。现需要将其中的某个单词替换成另一个单词，并输出替换之后的字符串。

【输入】第 1 行是包含多个单词的字符串 s；第 2 行是待替换的单词 a(长度 ≤ 100)；第 3 行是 a 将被替换的单词 b(长度 ≤ 100)。s,a,b 最前面和最后面都没有空格。

【输出】输出只有 1 行，将 s 中所有单词 a 替换成 b 之后的字符串。

【输入样例】

```
You want someone to help you
You
I
```

【输出样例】

```
I want someone to help you
```

```
You want someone to help you
You
I
I want someone to help you
```

5.6-7

```
#include<iostream>//7. 单词替换
#include<cstdio>
using namespace std;
string a,b,s[210];
string swap(string word)// 交换单词函数
{
      if(word==a)
            return b;
      else
            return word;
}
int main()
{
    int count=0;  // 单词数量
    char space;   // 单词间的空格
      do
    {
      cin>>s[count++];
      scanf("%c",&space);
      }while(space==' ');
      cin>>a>>b;
      for(int i=0;i<count;i++)
      {
            cout<<swap(s[i])<<" ";
      }
      return 0;
}
```

【题目描述】

请统计某个给定范围 [L,R] 的所有整数中，数字 2 出现的次数。

比如给定范围 [2,22]，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现 1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

【输入】

2 个正整数 L 和 R，之间一个空格隔开。（1 ≤ L ≤ R ≤ 100000）。

【输出】

数字 2 出现的次数。

【输入样例 1】	2 22	【输出样例 1】	6
【输入样例 2】	2 100	【输出样例 2】	20

5.6-8

```
#include<iostream>//8. 数字统计
#include<cstdio>
#include<cstring>
using namespace std;
char s[10];
int main()
{
	int l,r,ans=0;
	cin>>l>>r;
	for(int i=l;i<=r;i++)
	{
		sprintf(s,"%d",i);
		int len=strlen(s);
		for(int j=0;j<=len-1;j++)
		{
			if(s[j]=='2') ans++;
		}
	}
	cout<<ans;
	return 0;
}
```

【题目描述】

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零，例如输入 -380，反转后得到的新数为 -83。

【输入】输入共 1 行，一个整数 N。-1,000,000,000 ≤ N ≤ 1,000,000,000

【输出】输出共 1 行，一个整数，表示反转后的新数。

【输入样例】123

【输出样例】321

【输入输出样例 2】

输入：-380

输出：-83

5.6-9

```
#include<iostream>//9. 数字反转
#include<cstdio>
#include<cstring>
using namespace std;
char s[100],c[100];
int main()
{
	int n;
	cin>>n;
	sprintf(s,"%d",n);
	int l=strlen(s);
	for(int i=0;i<=l-1;i++)
	{
		c[l-i-1]=s[i];
	}
	if(n<0) cout<<"-";
	sscanf(c,"%d",&n);
	cout<<n;
	return 0;
}
```

按照字典序从小到大排序输出

输入：

2

England

China

输出：

China

England

5.6-10

```
#include<iostream>//10. 国家排序
#include<cstring>
#include<algorithm>
using namespace std;
string a[100001];
int main()
{
	int n;
	cin>>n;
	for(int i=0;i<n;i++)
	cin>>a[i];
	sort(a,a+n);
	for(int i=0;i<n;i++)
	{
		cout<<a[i]<<endl;
	}
	return 0;
}
```

问题描述：
在一个有 n 个学生的大班级中，存在两个学生生日相同的概率非常大，现在给出每个学生的名字，出生日月，试找出所有生日相同的学生
输入格式：
第一行为 n 个整数，表示有 n 个学生，n<=180；
此后 n 行，每行包含一个字符串和两个正整数，分别表示学生的名字（名字第一个字母大写，其余小写，不含空格，且长度小于 20），和出生月 m、出生日 d，1<=m<=12，1<=d<=31，名字、月、日之间用一个空格分隔
输出格式：
每组生日相同的学生，输出一行，其中前两个数字表示月和日，后面跟着所有在当天出生的学生的名字，数字、名字之间用一个空格分隔，对所有的输出，要求按日期从前到后的顺序输出，对生日相同的名字，按名字从短到长按序输出，长度相同按字典序输出，如果没有生日相同的学生，输出 None
输入样例：
6
Avril 3 2
Candy 4 5
Tim 3 2
Sufia 4 5
Lagrange 4 5
Bill 3 2
输出样例：
3 2 Tim Bill Avril
4 5 Candy Sufia Lagrange

5.6-11

```
#include<iostream>//11. 生日相同
#include<algorithm>
using namespace std;
struct student{
	string name,m,d;
}a[200];
int n;
```

```
bool comp(student x,student y)// 自定义比较函数
{
	if(x.m!=y.m) return x.m<y.m;
	if(x.d!=y.d) return x.d<y.d;
	if(x.name.size()!=y.name.size())
	{
		return x.name.size()<y.name.size();
		return x.name<y.name;
	}
}
int main()
{
	cin>>n;
	for(int i=1;i<=n;i++)
	{	cin>>a[i].name>>a[i].m>>a[i].d;	}

	sort(a+1,a+1+n,comp);
	bool flag=false;// 判断是否有生日相同
	for(int i=1;i<=n;i++)
	{
		if(a[i].m==a[i+1].m&&a[i].d==a[i+1].d)
		{
			cout<<a[i].m<<' '<<a[i].d<<' '<<a[i].name;
			while(a[i].m==a[i+1].m&&a[i].d==a[i+1].d)
			{
				flag=true;
				i++;
				cout<<' '<<a[i].name;
			}
			cout<<endl;
		}
	}
	if(!flag) cout<<"none";
	return 0;
}
```

有三个字符串 S,S1,S2，其中，S 长度不超过 300，S1 和 S2 的长度不超过 10。想检测 S1 和 S2 是否同时在 S 中出现，且 S1 位于 S2 的左边，并在 S 中互不交叉（即，S1 的右边界点在 S2 的左边界点的左侧）。计算满足上述条件的最大跨距（即，最大间隔距离：最右边的 S2 的起始点与最左边的 S1 的终止点之间的字符数目）。如果没有满足条件的 S1，S2 存在，则输出 -1。例如，S = "abcd123ab888efghij45ef67kl"，S1="ab"，S2="ef"，其中，S1 在 S 中出现了 2 次，S2 也在 S 中出现了 2 次，最大跨距为：18。

输入三个串：S，S1，S2，其间以逗号间隔（注意，S，S1，S2 中均不含逗号和空格）；
输出 :S1 和 S2 在 S 最大跨距；若在 S 中没有满足条件的 S1 和 S2，则输出 -1。

样例输入　abcd123ab888efghij45ef67kl,ab,ef

样例输出　18

5.6-12

```
#include<iostream>//12. 字符串最大间距
#include<algorithm>
using namespace std;
string b,a,c;
int l1,l2,dis1,dis2;
int main()
{
	getline(cin,b);
	l1=b.find(',');
	l2=b.rfind(',');
	a=b.substr(l1+1,l2-l1-1);
	c=b.substr(l2+1,b.size()-l2-1);
	b.erase(l1,b.size()-l1);
	dis1=b.find(a);
	dis2=b.rfind(c);
	if(dis1+l2-l1-2<dis2) cout<<dis2-dis1-(l2-l1-1);
	else cout<<"-1";
	return 0;
}
```

```
abcd123ab888efghij45ef67kl,ab,ef
18
```

试题描述

有两个由字符构成的环。请写一个程序，计算这两个字符环上最长连续公共字符串的长度。例如，字符串“ABCEFAGADEGKABUVKLM”的首尾连在一起，构成一个环；字符串“MADJKLUVKL”的首尾连在一起，构成一个另一个环；“UVKLMA”是这两个环的一个连续公共字符串。

输入格式

一行，包含两个字符串，分别对应一个字符环。这两个字符串之间用单个空格分开。字符串长度不超过 255，且不包含空格等空白符。

输出格式

输出一个整数，表示这两个字符环上最长公共字符串的长度。

样例输入

ABCEFAGADEGKABUVKLM MADJKLUVKL　（有多组测试数据）

样例输出

6

```
ABCEFAGADEGKABUVKLM MADJKLUVKL
6
```

5.6-13

```
#include<iostream>//13. 字符环
using namespace std;
string s1,s2,ss1,ss2,k;
int l1,l2,ans=-100000,f=-1;
int main()
{
	cin>>s1>>s2;
	ss1=s1+s1;
	ss2=s2+s2;
	l1=ss1.size();
	l2=ss2.size();// 模拟出首尾相连
	if(l1>l2)
	{
		swap(ss1,ss2);
		l1=ss1.size();
		l2=ss2.size();
	}
	for(int i=0;i<l1/2;i++)
	{
		for(int j=1;j<=l1/2;j++)
		{
			f=ss2.find(ss1.substr(i,j));// 在 ss2 中找 ss1 中 i 到 j 第一次出现的位置
			if(f!=-1)
			{
				if(j>ans)
				ans=j;
			}
		}
	}
	cout<<ans;
	return 0;
}
```

【题目描述】

R 国和S 国正陷入战火之中，双方都互派间谍，潜入对方内部，伺机行动。历尽艰险后，潜伏于 S 国的R 国间谍小C 终于摸清了S 国军用密码的编码规则：

1. S 国军方内部欲发送的原信息经过加密后在网络上发送，原信息的内容与加密后所得的内容均由大写字母‘A’-‘Z’构成（无空格等其他字符）。

2. S 国对于每个字母规定了对应的“密字”。加密的过程就是将原信息中的所有字母替换为其对应的“密字”。

3. 每个字母只对应一个唯一的“密字”，不同的字母对应不同的“密字”。“密字”可以和原字母相同。例如，若规定‘A’的密字为‘A’，‘B’的密字为‘C’（其他字母及密字略），则原信息“ABA”被加密为“ACA”。

现在，小 C 通过内线掌握了S 国网络上发送的一条加密信息及其对应的原信息。小C 希望能通过这条信息，破译S 国的军用密码。小C 的破译过程是这样的：扫描原信息，对于原信息中的字母 x（代表任一大写字母），找到其在加密信息中的对应大写字母 y，并认为在密码里 y 是 x 的密字。如此进行下去直到停止于如下的某个状态：

1. 所有信息扫描完毕，‘A’-‘Z’ 所有 26 个字母在原信息中均出现过并获得了相应的“密字”。

2. 所有信息扫描完毕，但发现存在某个（或某些）字母在原信息中没有出现。

3. 扫描中发现掌握的信息里有明显的自相矛盾或错误（违反S 国密码的编码规则）。例如某条信息“XYZ”被翻译为“ABA”就违反了“不同字母对应不同密字”的规则。

在小 C 忙得头昏脑涨之际，R 国司令部又发来电报，要求他翻译另外一条从S 国刚刚截取到的加密信息。现在请你帮助小C：通过内线掌握的信息，尝试破译密码。然后利用破译的密码，翻译电报中的加密信息。

【输入】

共3 行，每行为一个长度在 1 到100 之间的字符串。

第 1 行为小C 掌握的一条加密信息。

第 2 行为第1 行的加密信息所对应的原信息。

第 3 行为R 国司令部要求小C 翻译的加密信息。

输入数据保证所有字符串仅由大写字母‘A’-‘Z’构成，且第1 行长度与第2 行相等。

【输出】

共1 行。

若破译密码停止时出现 2，3 两种情况，请你输出“Failed”（不含引号，注意首字母大写，其它小写）。

否则请输出利用密码翻译电报中加密信息后得到的原信息。

【输入样例】

AA

AB

EOWIE

【输出样例】

Failed

【提示】

【输入输出样例 1 说明】

原信息中的字母‘A’和‘B’对应相同的密字，输出“Failed”。

【输入输出样例 2】

输入：

QWERTYUIOPLKJHGFDSAZXCVBN

ABCDEFGHIJKLMNOPQRSTUVWXY

DSLIEWO

输出：

Failed

【输入输出样例 3】

输入：

MSRTZCJKPFLQYVAWBINXUEDGHOOILSMIJFRCOPPQCEUNYDUMPP

YIZSDWAHLNOVFUCERKJXQMGTBPPKOIYKANZWPLLVWMQJFGQYLL

FLSO

输出：

NOIP

5.6-14

```
#include<bits/stdc++.h>//14. 潜伏者
using namespace std;
#define N 105
int main()
{
    char s_e[N], s_o[N], s_n[N];//s_e: 加密后字符串 s_o: 原字符 s_n: 待加密字符串
     char ori[128]={}, enc[128]={};//ori[i]:ASCII 码为 i 的加密字符的原字符 enc[i]:ASCII 码为 i 的原字符对应的加密字符 初值都为 '\0'
    cin>>s_e>>s_o>>s_n;
    int l1=strlen(s_e),l2=strlen(s_n),cn=0;//cn: 已经确定的加密关系的个数
    for(int i=0;i<l1;++i)
    {
        if(ori[s_e[i]]=='\0'&&enc[s_o[i]]=='\0')// 如果不存在 s_e[i] 对应的明文，同时不存在 s_o[i] 对应的密文
        {// 建立对应关系：明文 s_o[i] 对应密文 s_e[i]
            ori[s_e[i]]=s_o[i];
            enc[s_o[i]]=s_e[i];
            cn++;
        }
        else if(ori[s_e[i]]!=s_o[i]||enc[s_o[i]]!=s_e[i])
        {// 如果已有对应关系，且对应关系不为明文 s_o[i] 对应密文 s_e[i]
            cout<<"Failed";
            return 0;
        }
    }
    if(cn!=26)// 如果对应关系不足 26 对
    {
        cout<<"Failed";
        return 0;
    }
    for(int i=0;i<l2;i++)// 解密 s_n 字符串
        cout<<ori[s_n[i]];
    return 0;
}
```

现在输入一个字符串 s，还有 n 个旋转操作。每个操作有三个参数：a,b,c，意思是要把开始位置是 a，结束位置是 b 的这段字符串旋转 c 次。例如：字符串“abcdefg”，经过操作（2,5,2）后变成“abefcdg”。

输入：第一行，不包换空格的字符串 s，长度不超过 1000；第二行，一个整数 n(1<=n<=1000)，表示下面有 n 个旋转操作；接下来第 3 行到 n+3 行，每行三个整数 a,b(0<=a<=b<s),c(0<=c<10000)。输出：输出只有一行：将 s 旋转之后的字符串。

5.6-15

```
youwantsomeonetohelpyou

3
1 5 100
0 3 20
2 15 60


ynetonuwantsomeohelpyou
```

```
#include<iostream>//15. 旋转操作
using namespace std;
string s,s1;
int n;
int main()
{
	cin>>s;
	s1=s;
	cin>>n;
	for(int i=0;i<n;i++)
	{
		int a,b,c,len;
		cin>>a>>b>>c;
		len=b-a+1;
		c%=len;//c 大于字符串的长度，只要取余即可
		for(int j=a;j<=b;j++)// 备份下来到临时字符串 s1
		{
			s1[j]=s[j];
		}
		for(int j=a;j<=b;j++)
		{
			if(j+c>b) s[j+c-b]=s1[j];// 计算目标地址
			else s[j+c]=s1[j];
		}
	}
	cout<<s;
	return 0;
}
```

其他字符串试题（多选自 ccf）

5.7-1

```
#include<iostream>// 输入身份证，输出出生年月
using namespace std;
#include<string.h>
int main()
{
	int y,m,d;//year 年 ， month 月， day 日
	char id[18];
	cin>>id;
	cout<<id[6]<<id[7]<<id[8]<<id[9]<<id[10]<<id[11]<<id[12]<<id[13]<<endl;

	y=(id[6]-'0')*1000+(id[7]-'0')*100+(id[8]-'0')*10+(id[9]-'0');
	m=(id[10]-'0')*10+(id[11]-'0');
	d=(id[12]-'0')*10+(id[13]-'0');

	cout<<y<<" 年 "<<endl;
	cout<<m<<" 月 "<<endl;
	cout<<d<<" 日 ";

	return 0;
}
```

```
370283200011116060
20001111
2000 年
11 月
11 日
```

5.7-2

```
#include<iostream>// 输入身份证，输出年龄 本例题错误，缺少判断负数
#include<string.h>
using namespace std;
int main()
{
	int y,m,d;//year 年 ， month 月，day 日
	char id[18];
	cin>>id;
	cout<<id[6]<<id[7]<<id[8]<<id[9]<<id[10]<<id[11]<<id[12]<<id[13]<<endl;

	y=(id[6]-'0')*1000+(id[7]-'0')*100+(id[8]-'0')*10+(id[9]-'0');
	m=(id[10]-'0')*10+(id[11]-'0');
	d=(id[12]-'0')*10+(id[13]-'0');

	cout<<y<<" 年 "<<endl;
	cout<<m<<" 月 "<<endl;
	cout<<d<<" 日 "<<endl;

	cout<<" 年龄：出生以来距今：（按照 2022 年 06 月 21 日计算）"<<endl;
	cout<<2022-y<<" 岁 "<<endl;
	cout<<6-m<<" 月 "<<endl;
	cout<<21-d<<" 日 "<<endl;
	return 0;
}
```

月是个负数，需要修改程序。

```
370283200011116060
20001111
2000 年
11 月
11 日
年龄：出生以来距今：（按照
2022 年 06 月 21 日计算）
22 岁
-5 月
10 日
```

5.7-3

```
#include<iostream>// 输入身份证，输出年龄
#include<string.h>
using namespace std;
int main()
{
	int y,m,d,y2,m2,m3,d2;//year 年 ， month 月， day 日
	char id[18];
	cin>>id;
	cout<<id[6]<<id[7]<<id[8]<<id[9]<<id[10]<<id[11]<<id[12]<<id[13]<<endl;

	y=(id[6]-'0')*1000+(id[7]-'0')*100+(id[8]-'0')*10+(id[9]-'0');
	m=(id[10]-'0')*10+(id[11]-'0');
	d=(id[12]-'0')*10+(id[13]-'0');
	cout<<y<<" 年 "<<m<<" 月 "<<d<<" 日 "<<endl;

	cout<<" 年龄：出生以来距今：（按照 2022 年 06 月 10 日计算）"<<endl;
	y2=2022-y;
	m2=6-m;
	d2=10-d;
	cout<<" 第一次运算：  "<<y2<<" 年 "<<m2<<" 月 "<<d2<<" 日 "<<endl;

	if(d2<0)
	{
		m2=m2-1;
		d2=d2+30;
	}
	if(m2<0)
	{
		y2=y2-1;
		m2=m2+12;
	}
	cout<<" 第二次运算：  "<<y2<<" 年 "<<m2<<" 月 "<<d2<<" 日 ";
	return 0;
}
```

370283200011116060
20001111
2000 年 11 月 11 日
年龄：出生以来距今：（按照 2022 年 06 月 10 日计算）
第一次运算：22 年 -5 月 -1 日
第二次运算：21 年 6 月 29 日

370283200005206060
20000520
2000 年 5 月 20 日
年龄：出生以来距今：（按照 2022 年 06 月 10 日计算）
第一次运算：22 年 1 月 -10 日
第二次运算：22 年 0 月 20 日

370283200005056060
20000505
2000 年 5 月 5 日
年龄：出生以来距今：（按照 2022 年 06 月 10 日计算）
第一次运算：22 年 1 月 5 日
第二次运算：22 年 1 月 5 日

5.7-4

```
#include<iostream>// 输入身份证，看是否符合入学年龄
#include<string.h>
using namespace std;
int main()
{
	int y,m,d,y2,m2,m3,d2;//year 年 ， month 月， day 日
	char id[18];
	cin>>id;
	cout<<id[6]<<id[7]<<id[8]<<id[9]<<id[10]<<id[11]<<id[12]<<id[13]<<endl;

	y=(id[6]-'0')*1000+(id[7]-'0')*100+(id[8]-'0')*10+(id[9]-'0');
	m=(id[10]-'0')*10+(id[11]-'0');
	d=(id[12]-'0')*10+(id[13]-'0');
	cout<<y<<" 年 "<<m<<" 月 "<<d<<" 日 "<<endl;

	cout<<" 年龄：出生以来距今：（按照 2022 年 09 月 1 日入学计算）"<<endl;
	y2=2022-y;
	m2=9-m;
	d2=1-d;
	cout<<" 第一次运算：  "<<y2<<" 年 "<<m2<<" 月 "<<d2<<" 日 "<<endl;

	if(d2<0){    m2=m2-1;d2=d2+30; }
	if(m2<0){    y2=y2-1;m2=m2+12; }
	cout<<" 第二次运算：  "<<y2<<" 年 "<<m2<<" 月 "<<d2<<" 日 "<<endl;

	if(y2>=6) cout<<" 可以入学！ ";
	else    cout<<" 不够入学年龄 ";

	return 0;
}
```

370283201608266060
20160826
2016 年 8 月 26 日
年龄：出生以来距今：（按照 2022 年 09 月 1 日入学计算）
第一次运算：6 年 1 月 -25 日
第二次运算：6 年 0 月 5 日
可以入学！

370283201609026060
20160902
2016 年 9 月 2 日
年龄：出生以来距今：（按照 2022 年 09 月 1 日入学计算）
第一次运算：6 年 0 月 -1 日
第二次运算：5 年 11 月 29 日
不够入学年龄