

1. jhljx 学数组

Problem Description

终于到数组啦。。呵呵呵。。
jhljx 决定开始学习数组了。。
一天老师给了他一个字符串。。字符串吖。。能吃吗？
老师让你统计字符串中每个字符的个数。是不是很简单吖。

Input

输入多组数据。
每组数据为一个字符串，保证字符串的长度小于 1000。字符串中保证只有英文字母。

Output

输出字符串中出现的字符的个数。请按照先输出小写字母的个数，再输出大写字母的个数的顺序输出。每组数据间用一个空行隔开。详细请见样例。

Sample Input

AbccbA
abcdeBCCDE

Sample Output

```
# # #  
# # #  
[b][c][A]  
  
#  
# # # # # # # #  
[a][b][c][d][e][B][C][D][E]
```

Hint

C 语言使用字符串处理的库函数需要添加 `string.h` 头文件即`#include<string.h>`C 语言请用以下形式输入 `char a[1010];while(gets(a)){ int k=strlen(a);//计算字符串长度的函数,k 表示字符串的长度}`
C++语言使用字符串处理的库函数需要添加 `string.h` 头文件或者 `cstring` 头文件即`#include<cstring>`或者

```
#include<string.h>都可以 C++语言请用以下形式输入 char a[1010];while(cin>>a){    int k=strlen(a); //计算字符串长度的函数,k 表示字符串的长度    .....} 注意每组数据的输出结果之间有一个空行。未出现的字符不要输出。
```

解题思路：

本题与 c++ 课上所提到的竖着输出条形统计图的例题类似，同时还要按照先小写后大写的顺序输出，先按照 hint 中所给方法求出字符串长度 `length`，并运用循环，找出字符中 ASCII 码最大和最小的，同时建一个数组 `a` 统计每个字符输出次数，以便输出 “#” 时使用。

参考代码：

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char b;
    char c[10010];
    while(cin>>c)
    {
        int n;
        int x;
        int max=0;
        int min=200;
        int temp=0;
        int length=strlen(c); //计算字符串长度
        int a[200]={0};
        for(int i=0;i<=length-1;i++) //第一次循环
        {
            x=c[i];
            a[x]++;
            if(max<x) //找出 ASCII 码最大的
                max=x;
            if(temp<a[x]) //计算出现次数
                temp=a[x];
            if(min>x) //找出 ASCII 码最小的
                min=x;
        }
        for(int j=temp;j>=1;j--) //第二次循环
        {
            for(int m=97;m<=122;m++) //大写字符输出
            {
                if(a[m]>0)
```

```

        {
            if(a[m]>=j)
            {
                cout<<" # ";//存在一个输出一个 “#”

            }
            else
            {
                cout<<"   ";//否则输出空行

            }
        }
    }
    for(int m=min;m<=90;m++)//小写字符统计图的输出
    {
        if(a[m]>0)
        {
            if(a[m]>=j)
            {
                cout<<" # ";//存在一个输出一个 “#”
            }
            else
            {
                cout<<"   ";//否则输出空行
            }
        }
    }

    cout<<endl;//注意换行
}
for(int m=97;m<=max;m++)//图表坐标的输出（大写）
{
    if(a[m]>0)
    {
        b=m;
        cout<<"["<<b<<"]";
    }
}

for(int m=65;m<=90;m++)//图表坐标的输出（小写）
{
    if(a[m]>0)
    {
        b=m;

```

```
        cout<<"["<<b<<"]";
    }
}
cout<<endl<<endl;//注意每次输出一组数据结束要换两次行（题目要求）
}
```

2. 巫女的怀胎

Problem Description

若非技术，那便是才能。

比如 lxx 在算术方面具有卓越的才能。

已知 x 加上 x 的各个数字之和为 y ，则说明 x 和 y 是一对 cp。给出 $n(1 \leq n \leq 100000)$ 。

求 n 的最小 cp。无解输出 0。

Input

输入多组数据。（小于 100 组）

每组数据一行，为一个数 n 。($1 \leq n \leq 100000$)

Output

每组数据输出一行。为 n 的 cp。

Sample Input

216

121

Sample Output

198

0

解题思路：

从 1 一直找到 $n-1$ ，如果存在一个数 i ，使得 i 及其每位数的和等于 n ，则称 i 为其 cp，同时题目要求，必须找出最小 cp，因此，从 1 开始找按照分解 n 每位数的思路，将每位数拆开，

并与 i 加和，若满足则退出循环。

参考代码：

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    while(cin>>n)
    {
        int counter=0;
        for(int i=1;i<n;i++)//从 1 开始判断
        {

            if(i+i/100000+i/10000+(i/1000)%10+i%10+(i%100)/10+(i%1000)/100==n)//
            判断 i 是否为 n 的 cp
            {
                cout<<i<<endl;
                counter++;计数器加 1
                break;//若满足则退出循环
            }
            else
            {
                continue;//不满足进入下一次循环
            }
        }
        if(counter==0)
        {
            cout<<"0"<<endl;//如果 n 不存在 cp 的情况
        }
    }
}
```

3. 善恶的彼岸

Problem Description

You guys do not notice that we are gifted just for being humans.We are absolute predators, we do not even have any enemies.Maybe there are animals watching us and thinking that "Someday we will BEAT THEM DOWN!".

但是"Someday we will BEAT THEM DOWN!"这种格式会导致让人分不出是左引号还是右引号，所以我们需要把它变为“”。
一句话：把""变为“”。

Input

输入一篇文章。

Output

输出引号转换后的文章。

Sample Input

"To be or not to be," quoth the Bard, "that
is the question".

Sample Output

“To be or not to be,” quoth the Bard, “that
is the question”.

Hint

注意是将英文引号转化为中文引号。

解题思路：

重点是分开左，右引号讨论，因此要设置一判断变量来控制左右引号的输出（中文左右引号不同而英文的相同）。

参考代码：

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    char b;

    int counter=0;//判断变量
    while(scanf("%c",&b)!=EOF)
    {
```

```

        if(b!="")//如果 b 是普通字符
        {
            cout<<b;//直接输出
        }
        else//如果是引号
        {
            if(counter==0)//如果是左引号
            {
                cout<<"\"";
                counter=1;//判断变量改变
            }
            else//如果是右引号
            {
                cout<<"'";
                counter=0;//判断变量改变
            }
        }
    }
}

```

4. jhljx 分解质因数

Problem Description

jhljx 最近在学小学数学，老师教他分解质因数。也就是说给你一个数 n , 让你把它分解成若干个质数的乘积的形式。

Input

输入多组数据。

每组数据一行，为一个数 n 。($1 \leq n \leq 10^6$)

Output

每组数据输出一行。为一个表达式。表达式中的数字必须按从小到大的顺序排列。（详见样例）

Sample Input

11

9412

Sample Output

11
2*2*13*181

Hint

注意可能木有乘号哦。。

请用 `scanf` 和 `printf` 输入输出。

解题思路：

将一整数 n 分解成质因数相乘的形式，可利用判断 n 是否为质数的思路，因为最小的质数为 2，因此便令 i 从 2 开始循环，一直到根号 n ，每次递增 1。若 i 为 n 因数，则输出 i ，将 n/i 继续循环，直至根号 n 。其中需注意 “*” 的输出：只有当当时 n 不为 1 时才输出引号，因为无论哪个非质数进入循环，最后都会变为 1，而 1 不是质数，如果不这么做，会导致多输出一个 “*”。同时，在循环外，应再输出一个 $n(n!=1)$ ，保证质数的输出。当 n 等于 1 时单独讨论，输出 1。

参考代码：

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
int main()
{
    int n;
    while(scanf("%d",&n)!=EOF)
    {
        if(n==1)//当 n 等于 1
        {
            cout<<"1"<<endl;
        }
        else
        {
            for(int i=2;i<=sqrt(n);i++)//从 2 至根号 n 循环，找 n 的质因数（由小到大，可保证因数为质数）
            {
                while(n%i==0)//当 i 为 n 因数
                {
                    n/=i;//分解后的 n
```



```

        printf("%d", i); 输出质因数
        if(n!=1)
        {
            printf("*");//判断并输出 “*”
        }
    }
}
if(n!=1)
{
    printf("%d", n); //输出 n
}
printf("\n");//换行（题目要求）
}
}
}

```

5. jhljx 学排列组合

Problem Description

jhljx 最近又学习了一些排列组合的知识。比如什么隔板法，插空法神马的。。相信你们都会吧。。毕竟 College Entrance Examination 的数学考试中考过。。

于是 jhljx 决定来考考你。。你要是记不住的话去找高中数学老师吧。。~\(\(\cong \nabla \cong\)/~啦啦啦

你看我把题目的背景都告诉你了。。相信你能 AC 哟。。

快到圣诞节了。。于是松辰学妹送给他了一些糖果。。233

松辰学妹总共送了 n 种糖果。每种糖果的个数为 $a[1], a[2], a[3], \dots, a[n]$ 。jhljx 讨厌连续吃两块相同种类的糖。。所以他

会尽量避免吃到同种类的糖。

为啥捏？因为他是强迫症吖。。上一次上机他的强迫症不就犯了？请猛戳这里->[jhljx 的强迫症](#)

所以请你判断他是否会吃到同种类的糖。

Input

输入多组数据。

每组数据两行，第一行为一个数 $n(1 \leq n \leq 10000)$ ，第二行为 n 个数，每个数 $a[i]$ 表示 i 种类的糖共有多少个。(保证 $a[i]$ 在 int 范围内)

Output

每组数据输出一行。如果可以避免吃到同种类的糖，输出 Yes, 如果无法避免，输出 No。

Sample Input

3

4 1 1

Sample Output

No

解题思路：

用数学思路，当数目最多的一组比剩余的所有加和并加 1 还要大，则不能满足要求（插空法思路）

参考代码：

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    while(cin>>n)
    {
        int temp;
        int maxi=0;
        long long sum=0;
        for(int i=0;i<n;i++)
        {
            cin>>temp;
            if(temp>maxi)//找出 n 个数中的最大项
            {
                maxi=temp;
            }
            sum+=temp;//糖的总数目
        }
        if(maxi>(sum+1)/2)//判断是否满足（插空法）
        {
            cout<<"No"<<endl;
        }
        else
        {
            cout<<"YeS"<<endl;
        }
    }
}
```

