C++第六次上级解题报告

A **jhljx学数组**

解题思路：先统计所给的字符串中各自字母出现的次数,我是先开一个元素数目为123的数组b，然后输入的每一个字符都对应ASCII码表中的一个数字tem，然后令b[tem]增一，则最后b[i]代表的就是ASCII码表中序号为i的字符出现的次数。然后找到出现次数最多的字母出现的次数max，然后第一行到第max行输出#，其中第i行某个字母(假设其ascii码表中序号为j)需判断其在这一行是否需要输出#,判断方法是用b[j]+1-i，若大于零，则输出#，否则输出空格。

由于题目要求先输出小写字母再输出大写字母，而小写字母在ASCII中位于大写字母之后，故还需将小写字母都放在数组b靠前的元素。

参考代码：

#include <cstdio>

#include <string.h>

#include <iostream>

using namespace std;

int main()

{

char a[1001];

while(~scanf("%s",a))

{

int b[123]={};

int n=strlen(a);

int tem;

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

{

tem=a[i];

b[tem]++;

}

int max=0;

for (int i=0;i<123;i++)

{

if(b[i]>max)

max=b[i];

}

for (int i=0;i<26;i++)

swap(b[i],b[i+97]);

for (int i=max;i>=1;i--)

{

for (int j=0;j<123;j++)

{

if(b[j])

{

if(b[j]+1-i>0)

printf(" # ");

else

printf(" ");

}

}

printf("\n");

}

for (int i=0;i<123;i++)

{

if(b[i])

{

if(i<26)

printf("[%c]",i+97);

else

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

}

}

printf("\n\n");

}

}

B 巫女的怀胎

解题思路：

求n的CPy，由题目可知y必然小于n，该题利用遍历的方法，从1到n-1逐一比较，如果满足题目条件，则循环终止，输出y，否则输出0。注意求一个多为数n的十位数的方法为n/10%10,百位数字为n/100%10,千位数字为n/1000%10……以此类推。

参考代码：  
#include <iostream>

using namespace std;

int main()

{

int n;

while(cin>>n)

{

bool flag=1;

int i;

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

{

int a=i/100000%10;

int b=i/10000%10;

int c=i/1000%10;

int d=i/100%10;

int e=i/10%10;

int f=i%10;

int m=a+b+c+d+e+f;

if(i+m==n)

{

flag=0;

break;

}

}

if(flag)

cout<<0<<endl;

else

cout<<i<<endl;

}

}

C 善恶的彼岸

解题思路：

注意到中文的引号与英文的引号不同，并且中文的引号不是字符，中文的左引号与右引号不相同。但是文章中的引号一定是成对出现的。故可声明一个计数器，遇到一次英文的引号便另计数器＋1，每次输出中文的引号时根据计数器的奇偶性判断应该输出的是左引号还是右引号。

参考代码：

#include <cstdio>

int main()

{

char a;

int counter=1;

while(~scanf("%c",&a))

{

if(a=='"')

{

if (counter==1)

{

printf("“");

counter=2;

}

else

{

printf("”");

counter=1;

}

}

else

putchar(a);

}

}

D **jhljx分解质因数**

解题思路：1比较特殊，若n为1，则直接输出1.为了保存结果应开一个数组a，对于大于1的数n，令i从2开始递增，最多不大于n，若i为n的质因数，则将i保存到数组a中，令n/=i，然后保持i不变继续进行刚才的操作，知道遇到n无法整除i时，令i加一，继续进行之前的操作。这样不用担心会将一个非质数即合数保存到a中，因为如果n能够整除一个合数i，则n必定也能够整除i的质因数，故不会出现令n整除一个合数的情况。

此外，如果数组a中只保存了一个数，则说明n为质数，直接输出n，没有\*号。

参考代码：

#include <cstdio>

int main()

{

int n;

while(~scanf("%d",&n))

{

if(n==1)

{

printf("1\n");

continue;

}

int temp=n;

int a[1000]={};

int counter=0,i=2;

while(i<=temp)

{

if(n%i==0)

{

a[counter]=i;

n/=i;

counter++;

}

else

{

i++;

}

if(n==1)

break;

}

if(a[0]==temp)

{

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

continue;

}

else

{

int j;

for (j=0;a[j+1]>0;j++)

printf("%d\*",a[j]);

printf("%d\n",a[j]);

}

}

}

E jhljx学排列组合

解题思路：

先对输入的数组进行排序，然后比较数组前n-1项之和与第n项减一的大小，如果前者大或两者相同，则一定可以不出现连续两颗相同的糖。如果后者大，则必定会出现至少两颗连续且相同的糖。

原理：由于数组已经排好序，则应先将b[n-2]插入到b[n-1]中，然后再将b[n-3]插入到前两者所形成的队列中，然后将b[n-4]再插入到前三者所形成的队列中，以此类推。如果b[0]+b[1]+……+b[n-1]<b[n-1]-1,即第n种糖两两之间不能全部插入别的种类的糖，则可知至少有两个第n种的糖要连续放在一起，故不满足题意；如果b[0]+b[1]+……+b[n-1]>=b[n-1]-1,则首先可将第n种糖两两之间全部插入第三种糖，并且第n-1中糖也一定全部放完了。剩余的第n-2种糖由于数目小于等于第n-1种糖的数目，故一定要么插在第n种糖之间，要么插在第n-1种糖之间，抑或是第n中与第n-1种糖之间。依次类推，所有种类的糖一定都可以间隔开来。即这是一个充分必要条件。

注意在对数组前n-1项求和是可能超过int范围，故声明数组时应声明为long long int类型的。

参考代码：

#include <iostream>

using namespace std;

void abc(int,int ,long long int []);

int main()

{

int n;

while(cin>>n)

{

long long int a[10002];

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

cin>>a[i];

abc(0,n-1,a);

long long int ans=0;

for (int i=0;i<n-1;i++)

ans+=a[i];

if(ans<a[n-1]-1)

cout<<"No"<<endl;

else

cout<<"YeS"<<endl;

}

}

void abc(int p,int r,long long int a[])

{

if(p<r)

{

int i=p-1;

for (int j=p;j<r;j++)

{

if(a[j]<a[r])

{

i++;

swap(a[i],a[j]);

}

}

i++;

swap(a[i],a[r]);

abc(p,i-1,a);

abc(i+1,r,a);

}

}