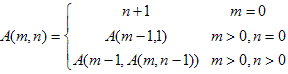
# 2014级C++第六次练习赛

巩毅琛 14211072

抱歉这次解题报告很水。

## A jhljx又来学函数

### Problem Description

jhljx最近很爱学习吖。。咕~~(╯﹏╰)b。。他决定把函数重新温习一下。。233  
于是他找到了这样一个函数。  
  
这个函数好神奇吖。。他给了你一个m和n,让你来求出A(m,n)的值。

### Input

输入多组数据。  
每组数据一行。包括两个数m和n。m可以取0,1,2,3。当m取0,1,2时，0<=n<=1000000。当m取3时，0<=n<=24。

### Output

每组数据输出一行，为一个数，即A(m,n)。

### Sample Input

1 3  
2 4

### Sample Output

5  
11

### Hint

/\*这是你们看不到的注释  
这道题可以用递归来水。但是得不到全分哦。。要想拿满分，去百度一下把。。啦啦啦~~~  
我不会说网上会有计算公式哦。。

\*/

解题思路：阿克曼函数。确实经典递归，但是由于它的增长速度太快,用递归是很容易溢出的。所以非递归实现。。。这里的代码是从网上找到的可是原文没有解释。网友是把m为0,1,2,3的情况都找出了规律，每递归一步m肯定减1，直到m=0。0,1,2的规律比较好找，但3的规律比较麻烦还没纸上推出来，但真的可以找到规律，由于时间关系我只好这样大体说说。

参考代码：

#include <stdio.h>

int main()

{

int value[25];

int i,m,n;

value[0] = 5;

for (i=1; i<25; i++)

{

value[i] = value[i - 1] \* 2 + 3;

}

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

{

switch(m)

{

case 0: printf("%d\n", n + 1);

break;

case 1: printf("%d\n", n + 2);

break;

case 2: printf("%d\n", 3 + 2 \* n);

break;

case 3: printf("%d\n", value[n]);

break;

default:break;

}

}

return 0;

}

## B Results

### Description

宋友老师给了Thor一些同学的成绩，让他把柱状图打出来。现在他来拉你做壮丁了。  
具体请看下面：的样例。

### Input

多组数据。  
每组数据有2行。  
第一行有1个整数n(1<=n<=1000)。  
第二行有n个整数A[i](0<=A[i]<=100)表示每个人的成绩。

### Output

对于每组数据  
输出柱状图，用'\*'输出。  
每次输出场宽为4。

### Sample Input

4

80 80 90 1

5

60 65 64 63 62

### Sample Output

\*

\* \* \*

1 80 90

\* \* \* \* \*

60 62 63 64 65

### Hint

注意每组输出最后有一个换行。

解题思路：将字符二维数组用作坐标系来做。首先数组所有项初始化为空格，然后在输入成绩时，用另一个数组来统计分别得到相应成绩的人数，然后用这个数组的统计结果，将原先那个字符数组相应成绩对应的列上赋值相应人数的星号，该成绩有多少个人就在该列的几行上赋值星号。然后到了输出阶段，输出每个字付前场宽都是4，先找到统计的人数中的最大值m，所以在输出星号部分输出m行，这样行数及具体哪几行可以确定。要确定哪几列，则就是统计人数不为0的相应分数对应的字符数组的列，于是列也确定。然后一行一行输出即可。最后一行，输出统计人数不为0的相应分数即可。

参考代码：

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

char tu[1001][101];

int sc[101];

int main()

{

int n;

while(cin>>n)

{

memset(sc,0,sizeof(sc));

for(int i=0;i<1001;i++)

for(int j=0;j<101;j++)

tu[i][j]=' ';

int shuru;

for(int i=1;i<=n;i++)//统计各分数人数

{

cin>>shuru;

sc[shuru]++;

}

for(int i=0;i<101;i++)//填充绘制柱状图

{

for(int j=0;j<sc[i];j++)

{

tu[j][i]='\*';

}

}

int m=-1;

for( int i=0;i<101;i++)//求统计人数的最大值

{

if(sc[i]!=0&&sc[i]>m)

m=sc[i];

}

for(int i=0;i<m;i++)//柱状图部分

{

for(int j=0;j<101;j++)

{

if(sc[j]!=0)

{

cout<<setw(4)<<tu[m-i-1][j];

}

}

cout<<endl;

}

for(int i=0;i<101;i++)//最后一行

{

if(sc[i]!=0)

cout<<setw(4)<<i;

}

cout<<endl<<endl;

}

}

## C The Super N

**题目描述**

马上就要放假了，Alice和Bob觉得非常无聊，所以决定一起玩一个游戏。由于Alice和Bob都不是特别聪明，所以游戏规则非常简单：

Alice给出一个字母C（小写'a'-'z'）和一个数字n，而Bob则要以这个字母为起始，按招Alice的要求画出一个大小为n的字母'N'。

但是Alice没有告诉Bob他规则的具体内容，只给了Bob几个例子：

比如以'a'为起始字母，大小为3的字母N张这个样子：

a e

bdf

c g

再比如以'z'为起始字母，大小为5的字母N张这个样子：

z h

ae i

b f j

c gk

d l

Bob看了之后非常费解，完全不知道到底要怎么画这个‘N‘.但是游戏已经开始了，所以，Bob就只能求助于智商更高的你了。

如果你能帮助Bob解决这个问题，Alice还会赐予你**‘The Super N’**的称号！

**输入格式**

第一行一个正整数T(T<=100)，表示测试数据组数。

每组测试数据一行，每行一个小写字母C（'a'-'z'）和一个数字n（3<=n<=10）

**输出格式**

每组测试数据输出n行（因为要输出的字母‘N’是n行~跟Alice关系很好的NExPlain这么提示着你），按题目要求输出。

**样例输入**

2

a 3

q 5

**样例输出**

a e

bdf

c g

q y

rv z

s w a

t xb

u c

**Source**

BUAA\_NExPlain

解题思路：使用二维数组作为坐标系来做即可。我建立的是整数数组，所以存入数组的是输入字符的ASCII码。在为数组填充数据时，即完善坐标系中的内容时，分了三部分，1第一列是从上到下的顺序，2中间部分是左上到右下的顺序，3最后一列也是从上到下。于是以这种顺序将输入字符的ACSII码加一递增然后逐个输入到数组对应的位置上。

在输出阶段，先判断数组中某个数是不是0，如果是则输出空格，如果不是，则将该数减96，（可以看做1就是a，2就是b，3就是c。。。），然后模26取余，（存在大于26的情况，可只有26个字母，如果是27那就代表着a了，其中余数为0的话代表z要特殊处理），然后把余数加上96后输出该数对应的char字符即可。

参考代码：

#include <iostream>

using namespace std;

const int maxn=100;

int main()

{

int T;

while(cin>>T)

{

char c;int n;

for(int i=1;i<=T;i++)

{

int a[maxn][maxn]={0};

cin>>c>>n;

int C=int(c);

a[1][1]=C;

for(int h=2;h<=n;h++)//第一部分

{

a[h][1]=++C;

}

for(int h=2;h<=n-1;h++)//第二部分

{

a[h][h]=++C;

}

for(int h=1;h<=n;h++)//第三部分

{

C++;

a[h][n]=C;

}

for(int p=1;p<=n;p++)//输出

{

for(int q=1;q<=n;q++)

{

if(a[p][q]==0)

cout<<" ";

else

{

if((a[p][q]-96)%26==0)//取余数目前就是为了在z和a衔接好

cout<<"z";

else

{

cout<<char((a[p][q]-96)%26+96);

}

}

}

cout<<endl;

}

}

}

}

## E Choice II

解题思路：首先想到的是三个嵌套的for语句，但是这样1000^3就超时了。所以为了减小时间复杂度使用二分法。课堂上老师讲的是用二分法从一个已经排好序的数列查找某个数，那这里要查找的是，在第一个第二个数确定的情况下，哪一个数作为第三个数与前两个数相加等于或刚好大于s，只是对于这道题在查找的过程中有一个累加的计算。

就是说在排好序后（用的树荫学姐习惯的冒泡排序），在一个第一个第二个数确定好的情况里，取第三个数为第二个数下一个数和最后一个数中间的那个数，判断这个中间数加上前两个数是否符合条件，若符合则符合条件的个数需累加上这个中间数后面所有数的个数，同时把新中间数设置为原中间数上一个数与第二个数下一个数之间的中间数，并继续判断，若不符合，则将中间数设为最后一个数和原先中间数下一个数之间的中间数，并继续判断。直到找到那个使和等于或刚好大于s的第三个数，这时累加也完成了结果正是所有可能的结果。

参考代码：

#include <iostream>

#include <cstring>

using namespace std;

int a[1001];

int main()

{

int n, s;

while(cin>>n>>s)

{

memset(a, 0, sizeof(a));

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

cin>>a[i];

for(int i=1;i<=n;i++)//排序

{

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

{

if(a[i]>a[j])

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

}

}

int t=0;

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

{

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

{

int l=j+1,r=n,m;

while(l<=r)//二分法找到那个特殊值时也进行着符合条件取法个数的累加

{

m=(l+r)/2;

if(a[i]+a[j]+a[m]>s)

{

t=t+r-m+1;

r=m-1;

}

else

l=m+1;

}

}

}

cout<<t<<endl;

}

}