**第三届蓝桥桥杯预赛真题及详细答案**

1. 微生物增殖

假设有两种微生物 X 和 Y

X出生后每隔3分钟分裂一次（数目加倍），Y出生后每隔2分钟分裂一次（数目加倍）。 一个新出生的X，半分钟之后吃掉1个Y，并且，从此开始，每隔1分钟吃1个Y。

现在已知有新出生的 X=10, Y=89，求60分钟后Y的数目。

如果X=10，Y=90 呢？

本题的要求就是写出这两种初始条件下，60分钟后Y的数目。

题目的结果令你震惊吗？这不是简单的数字游戏！真实的生物圈有着同样脆弱的性质！也许因为你消灭的那只 Y 就是最终导致 Y 种群灭绝的最后一根稻草！

请忍住悲伤，把答案写在“解答.txt”中，不要写在这里！

答案：0 和 94371840

(此题比较简单就不写程序了，以下均有程序)

二、古堡算式

福尔摩斯到某古堡探险，看到门上写着一个奇怪的算式：

ABCDE \* ? = EDCBA

他对华生说：“ABCDE应该代表不同的数字，问号也代表某个数字！”

华生：“我猜也是！”

于是，两人沉默了好久，还是没有算出合适的结果来。

请你利用计算机的优势，找到破解的答案。

把 ABCDE 所代表的数字写出来。

答案写在“解答.txt”中，不要写在这里！

参考答案：21978

参考程序：

#include<stdio.h>

void main(){

int k,t1 ,t;

for(k=10234;k<50000;k++){

bool flag[10]={0};

t=k;

t1=0;

while(t){

if(flag[t%10])

break ;

flag[t%10] = true ;

t1 = t1\*10 + t%10 ;

t /= 10 ;

}

if( t==0 && t1%k==0){

printf("%d \* %d = %d\n",k,t1/k,t1);

}

}

}

三、比酒量

有一群海盗（不多于20人），在船上比拼酒量。过程如下：打开一瓶酒，所有在场的人平分喝下，有几个人倒下了。再打开一瓶酒平分，又有倒下的，再次重复...... 直到开了第4瓶酒，坐着的已经所剩无几，海盗船长也在其中。当第4瓶酒平分喝下后，大家都倒下了。

等船长醒来，发现海盗船搁浅了。他在航海日志中写到：“......昨天，我正好喝了一瓶.......奉劝大家，开船不喝酒，喝酒别开船......”

请你根据这些信息，推断开始有多少人，每一轮喝下来还剩多少人。

如果有多个可能的答案，请列出所有答案，每个答案占一行。

格式是：人数,人数,...

例如,有一种可能是：20,5,4,2,0

答案写在“解答.txt”中，不要写在这里！

参考答案：

12 6 4 2 0

15 10 3 2 0

18 9 3 2 0

20 5 4 2 0

参考程序：

#include<stdio.h>

void f(int a[],int n){

int i ;

if(n==3){

int x,y;

x = a[0] \* a[1] + a[0] \* a[2] + a[1] \* a[2] ;

y = a[0] \* a[1] \* a[2] ;

x = y - x ;

if( x>0 && y%x==0 && y/x<a[2] ){

printf("%d %d %d %d 0\n",a[0],a[1],a[2],y/x);

}

return;

}

i = a[n-1] - 1 ;

while(i){

a[n] = i ;

f(a,n+1);

i-- ;

}

}

void main(){

int a[3]={0};

for(int i=4;i<=20;i++){

a[0] = i ;

f(a,1);

}

}

四、奇怪的比赛

某电视台举办了低碳生活大奖赛。题目的计分规则相当奇怪：

每位选手需要回答10个问题（其编号为1到10），越后面越有难度。答对的，当前分数翻倍；答错了则扣掉与题号相同的分数（选手必须回答问题，不回答按错误处理）。

每位选手都有一个起步的分数为10分。

某获胜选手最终得分刚好是100分，如果不让你看比赛过程，你能推断出他（她）哪个题目答对了，哪个题目答错了吗？

如果把答对的记为1，答错的记为0，则10个题目的回答情况可以用仅含有1和0的串来表示。例如：0010110011 就是可能的情况。

你的任务是算出所有可能情况。每个答案占一行。

答案写在“解答.txt”中，不要写在这里！

参考答案：

1011010000

0111010000

0010110011

参考程序：

#include<stdio.h>

//score为完成第n题之后的分数

void f(char s[] , int n , int score){

if(n==0){

if(score==10)

puts(s);

return;

}

//第n题答错

s[n-1] = '0' ;

f( s , n-1 , score+n );

//第n题答对

if(score%2==0){

s[n-1] = '1' ;

f(s,n-1,score/2);

}

}

void main(){

char s[10];

s[10] = '\0' ;

f(s,10,100);

}

五、转方阵

对一个方阵转置，就是把原来的行号变列号，原来的列号变行号

例如，如下的方阵：

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

转置后变为：

1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

但，如果是对该方阵顺时针旋转（不是转置），却是如下结果：

13 9 5 1

14 10 6 2

15 11 7 3

16 12 8 4

下面的代码实现的功能就是要把一个方阵顺时针旋转。

void rotate(int\* x, int rank)

{

int\* y = (int\*)malloc(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_); // 填空

for(int i=0; i<rank \* rank; i++)

{

y[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] = x[i]; // 填空

}

for(i=0; i<rank\*rank; i++)

{

x[i] = y[i];

}

free(y);

}

int main(int argc, char\* argv[])

{

int x[4][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};

int rank = 4;

rotate(&x[0][0], rank);

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

{

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

{

printf("%4d", x[i][j]);

}

printf("\n");

}

return 0;

}

请分析代码逻辑，并推测划线处的代码。

答案写在 “解答.txt” 文件中

注意：只写划线处应该填的内容，划线前后的内容不要抄写。

参考答案：

rank\*rank\*sizeof(int)

(i%rank)\*rank + rank-i/rank-1

六、大数乘法

对于32位字长的机器，大约超过20亿，用int类型就无法表示了，我们可以选择int64类型，但无论怎样扩展，固定的整数类型总是有表达的极限！如果对超级大整数进行精确运算呢？一个简单的办法是：仅仅使用现有类型，但是把大整数的运算化解为若干小整数的运算，即所谓：“分块法”。

如图【1.jpg】表示了分块乘法的原理。可以把大数分成多段（此处为2段）小数，然后用小数的多次运算组合表示一个大数。可以根据int的承载能力规定小块的大小，比如要把int分成2段，则小块可取10000为上限值。注意，小块在进行纵向累加后，需要进行进位校正。

以下代码示意了分块乘法的原理（乘数、被乘数都分为2段）。

void bigmul(int x, int y, int r[])

{

int base = 10000;

int x2 = x / base;

int x1 = x % base;

int y2 = y / base;

int y1 = y % base;

int n1 = x1 \* y1;

int n2 = x1 \* y2;

int n3 = x2 \* y1;

int n4 = x2 \* y2;

r[3] = n1 % base;

r[2] = n1 / base + n2 % base + n3 % base;

r[1] = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // 填空

r[0] = n4 / base;

r[1] += \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // 填空

r[2] = r[2] % base;

r[0] += r[1] / base;

r[1] = r[1] % base;

}

int main(int argc, char\* argv[])

{

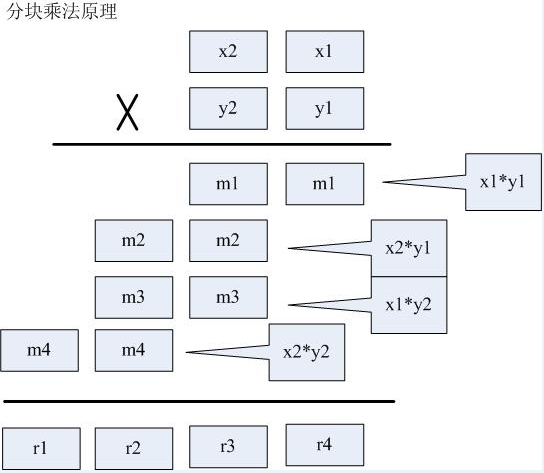
int x[] = {0,0,0,0};

bigmul(87654321, 12345678, x);

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

return 0;

}



请分析代码逻辑，并推测划线处的代码。

答案写在 “解答.txt” 文件中

注意：只写划线处应该填的内容，划线前后的内容不要抄写。

参考答案:

n2/base + n3/base + n4%base

r[2]/base

七、放棋子

今有 6 x 6 的棋盘格。其中某些格子已经预先放好了棋子。现在要再放上去一些，使得：每行每列都正好有3颗棋子。我们希望推算出所有可能的放法。下面的代码就实现了这个功能。

初始数组中，“1”表示放有棋子，“0”表示空白。

int N = 0;

bool CheckStoneNum(int x[][6])

{

for(int k=0; k<6; k++)

{

int NumRow = 0;

int NumCol = 0;

for(int i=0; i<6; i++)

{

if(x[k][i]) NumRow++;

if(x[i][k]) NumCol++;

}

if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) return false; // 填空

}

return true;

}

int GetRowStoneNum(int x[][6], int r)

{

int sum = 0;

for(int i=0; i<6; i++) if(x[r][i]) sum++;

return sum;

}

int GetColStoneNum(int x[][6], int c)

{

int sum = 0;

for(int i=0; i<6; i++) if(x[i][c]) sum++;

return sum;

}

void show(int x[][6])

{

for(int i=0; i<6; i++)

{

for(int j=0; j<6; j++) printf("%2d", x[i][j]);

printf("\n");

}

printf("\n");

}

void f(int x[][6], int r, int c);

void GoNext(int x[][6], int r, int c)

{

if(c<6)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // 填空

else

f(x, r+1, 0);

}

void f(int x[][6], int r, int c)

{

if(r==6)

{

if(CheckStoneNum(x))

{

N++;

show(x);

}

return;

}

if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_) // 已经放有了棋子

{

GoNext(x,r,c);

return;

}

int rr = GetRowStoneNum(x,r);

int cc = GetColStoneNum(x,c);

if(cc>=3) // 本列已满

GoNext(x,r,c);

else if(rr>=3) // 本行已满

f(x, r+1, 0);

else

{

x[r][c] = 1;

GoNext(x,r,c);

x[r][c] = 0;

if(!(3-rr >= 6-c || 3-cc >= 6-r)) // 本行或本列严重缺子，则本格不能空着！

GoNext(x,r,c);

}

}

int main(int argc, char\* argv[])

{

int x[6][6] = {

{1,0,0,0,0,0},

{0,0,1,0,1,0},

{0,0,1,1,0,1},

{0,1,0,0,1,0},

{0,0,0,1,0,0},

{1,0,1,0,0,1}

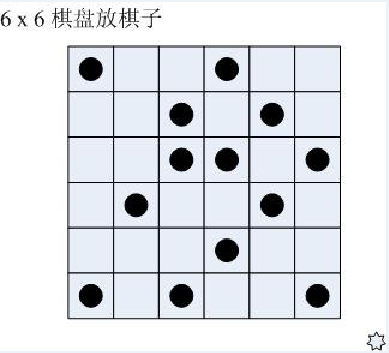
};

f(x, 0, 0);

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

return 0;

}



请分析代码逻辑，并推测划线处的代码。

答案写在 “解答.txt” 文件中

注意：只写划线处应该填的内容，划线前后的内容不要抄写。

参考答案：

NumRow!=3 || NumCol!=3

f(x, r, c+1)

x[r][c]

八、密码发生器

在对银行账户等重要权限设置密码的时候，我们常常遇到这样的烦恼：如果为了好记用生日吧，容易被破解，不安全；如果设置不好记的密码，又担心自己也会忘记；如果写在纸上，担心纸张被别人发现或弄丢了...

这个程序的任务就是把一串拼音字母转换为6位数字（密码）。我们可以使用任何好记的拼音串(比如名字，王喜明，就写：wangximing)作为输入，程序输出6位数字。

变换的过程如下：

第一步. 把字符串6个一组折叠起来，比如wangximing则变为：

wangxi

ming

第二步. 把所有垂直在同一个位置的字符的ascii码值相加，得出6个数字，如上面的例子，则得出：

228 202 220 206 120 105

第三步. 再把每个数字“缩位”处理：就是把每个位的数字相加，得出的数字如果不是一位数字，就再缩位，直到变成一位数字为止。例如: 228 => 2+2+8=12 => 1+2=3

上面的数字缩位后变为：344836, 这就是程序最终的输出结果！

要求程序从标准输入接收数据，在标准输出上输出结果。

输入格式为：第一行是一个整数n（<100），表示下边有多少输入行，接下来是n行字符串，就是等待变换的字符串。

输出格式为：n行变换后的6位密码。

例如，输入：

5

zhangfeng

wangximing

jiujingfazi

woaibeijingtiananmen

haohaoxuexi

则输出：

772243

344836

297332

716652

875843

注意：

请仔细调试！您的程序只有能运行出正确结果的时候才有机会得分！

在评卷时使用的输入数据与试卷中给出的实例数据可能是不同的。

请把所有函数写在同一个文件中，调试好后，存入与【考生文件夹】下对应题号的“解答.txt”中即可。

相关的工程文件不要拷入。

源代码中不能能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的API。

允许使用STL类库，但不能使用MFC或ATL等非ANSI c++标准的类库。例如，不能使用CString类型（属于MFC类库）。

参考程序：

#include<stdio.h>

#include<string.h>

int reduction(int m){

int k=0;

if(m<10)

return m;

while(m){

k += m%10 ;

m /= 10 ;

}

return reduction(k);

}

void f(char s[],char x[]){

int i,j,n,k;

n = strlen(s) ;

for(i=0;i<6;i++){

for(k=0,j=i;j<n;j+=6)

k+=s[j];

x[i] = reduction(k) + '0' ;

}

x[6] = '\n' ;

}

void main(){

char s[100]={"wangximing"};

char x[10000]={""};

int len = 0 , n ;

scanf("%d",&n);

while(n){

n--;

scanf("%s",s);

f(s,x+len);

len += 7 ;

}

x[len-1]='\0' ;

puts(x);

}

九、夺冠概率

足球比赛具有一定程度的偶然性，弱队也有战胜强队的可能。

假设有甲、乙、丙、丁四个球队。根据他们过去比赛的成绩，得出每个队与另一个队对阵时取胜的概率表:

甲 乙 丙 丁

甲 - 0.1 0.3 0.5

乙 0.9 - 0.7 0.4

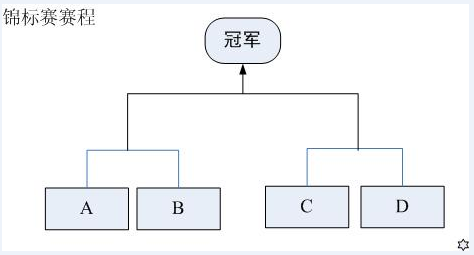
丙 0.7 0.3 - 0.2

丁 0.5 0.6 0.8 -

数据含义：甲对乙的取胜概率为0.1，丙对乙的胜率为0.3，...

现在要举行一次锦标赛。双方抽签，分两个组比，获胜的两个队再争夺冠军。（参见【1.jpg】）

请你进行10万次模拟，计算出甲队夺冠的概率。



注意：

请仔细调试！您的程序只有能运行出正确结果的时候才有机会得分！

在评卷时使用的输入数据与试卷中给出的实例数据可能是不同的。

请把所有函数写在同一个文件中，调试好后，存入与【考生文件夹】下对应题号的“解答.txt”中即可。

相关的工程文件不要拷入。

源代码中不能能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的API。

允许使用STL类库，但不能使用MFC或ATL等非ANSI c++标准的类库。例如，不能使用CString类型（属于MFC类库）。

参考程序：

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

void main(){

int pk[4][4] = {

{0,1,3,5},

{9,0,7,4},

{7,3,0,2},

{5,6,8,0}};

int i,a,a1,b,b1,k,count=0;

srand(time(NULL));

for(i=0;i<100000;i++){

a1 = rand()%3 + 1 ;//a的对手

if(rand()%10<pk[0][a1]){//a胜

switch(a1){

case 1:

b = 2 ;

b1 = 3 ;

break;

case 2:

b = 1 ;

b1 = 3 ;

break;

default:

b = 1 ;

b1 = 2 ;

}

if(rand()%10<pk[b][b1]){//b胜

a1 = b ;

}

else{//b1胜

a1 = b1 ;

}

if(rand()%10<pk[0][a1]){//a胜

count++;

}

}

}

printf("%f\n",count\*1.0/100000);

}

十、取球游戏

今盒子里有n个小球，A、B两人轮流从盒中取球，每个人都可以看到另一个人取了多少个，也可以看到盒中还剩下多少个，并且两人都很聪明，不会做出错误的判断。

我们约定：

每个人从盒子中取出的球的数目必须是：1，3，7或者8个。

轮到某一方取球时不能弃权！

A先取球，然后双方交替取球，直到取完。

被迫拿到最后一个球的一方为负方（输方）

请编程确定出在双方都不判断失误的情况下，对于特定的初始球数，A是否能赢？

程序运行时，从标准输入获得数据，其格式如下：

先是一个整数n(n<100)，表示接下来有n个整数。然后是n个整数，每个占一行（整数<10000），表示初始球数。

程序则输出n行，表示A的输赢情况（输为0，赢为1）。

例如，用户输入：

４

１

２

10

18

则程序应该输出：

0

1

1

0

注意：

请仔细调试！您的程序只有能运行出正确结果的时候才有机会得分！

在评卷时使用的输入数据与试卷中给出的实例数据可能是不同的。

请把所有函数写在同一个文件中，调试好后，存入与【考生文件夹】下对应题号的“解答.txt”中即可。

相关的工程文件不要拷入。

源代码中不能能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的API。

允许使用STL类库，但不能使用MFC或ATL等非ANSI c++标准的类库。例如，不能使用CString类型（属于MFC类库）。

参考程序：

#include<stdio.h>

void main(){

int a[100],n,max;

int b[]={1,3,7,8};

bool flag[10001]={0};

int i,j;

max = 0 ;

scanf("%d",&n);

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

scanf("%d",a+i);

if(a[i]>max)

max = a[i] ;

}

for(i=2;i<=max;i++){

for(j=0 ; j<4 && b[j]<i ; j++ ){

if(flag[i-b[j]]==0){

flag[i] = 1 ;

break;

}

}

}

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

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

}