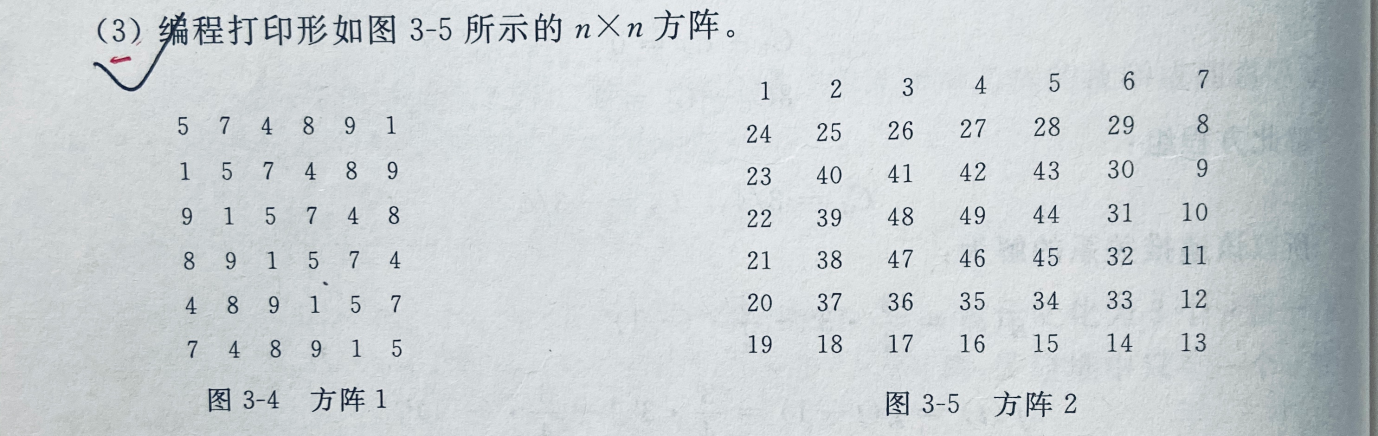
**第1次作业: 3 11 13 14**



3.思路：首先观察该矩阵，发现其规律为自左向右自上向下依次增大，利用指针变化，确定数组位置，一次由外到内依次打印数字。

伪代码：void fun( ){

}

源程序

#include<stdio.h>

int a[7][7];

void fun(int n)

{

int i,j;

int m=1;

int s=0,p=n;

while(m<=n\*n)//m是1~n\*n之间的数

{

for(i=s;i<p;i++)

{//顺时针向右递增

if(i==s)

{

for(j=s;j<p;j++)

{

a[i][j]=m++;

}

}//顺时针向下递增

if(i!=s)

{

j=p-1;

a[i][j]=m++;

}

if(i==(p-1))

{//顺时针向左递增

for(j=p-2;j>=s;j--)

{

a[i][j]=m++;

}

//顺时针向上递增

for(i=p-2,j=s;i>s;i--)

{

a[i][j]=m++;

}

i=p-1;

}

}

s++;//顺时针转一圈后，a[i][j]初始值为a[1][1]，转两圈后初始值为a[2][2],以此类推

p--;//转一圈后，i,j最大值分别变为n-2,转两圈后变为n-3,以此类推

}

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

{

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

{

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

}

printf("\n");

}

}

int main()

{

int i,j;

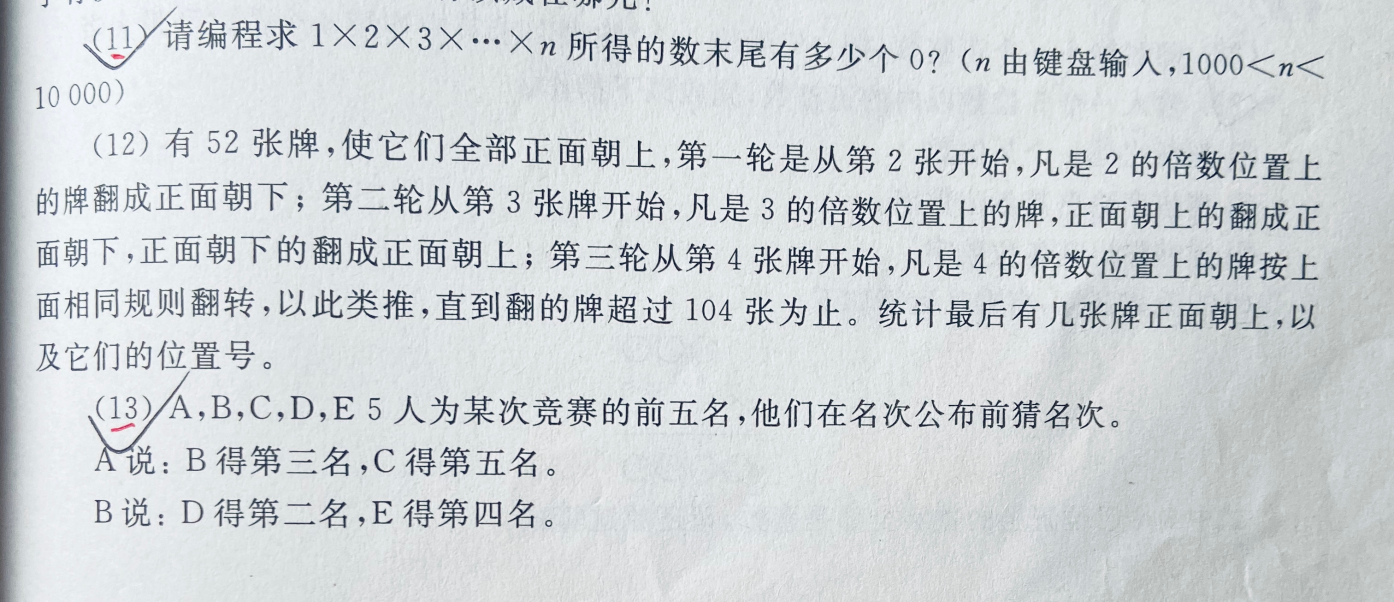
for(i=0;i<7;i++)

for(j=0;j<7;j++)

a[i][j]=0;

fun(7);

return 0;

}

11.思路：由1000<n<10000可知该数不可直接算出，思考0出现的条件可知，当此式可分解出2和5时即可出现0，所以将其因式分解，寻找其中的2和5的即可得出答案

伪代码：void findZero() {

input(n);

fiveCount = 0;

for i = 2 to n do

{

tem = i;

while (tem % 5 == 0) do

{ tem /= 5;

fiveCount++;}

}

print(fiveCount);

}

源程序：#include <stdio.h>

void findZero()

int n;

printf("请输入n\n");

scanf("%d", &n);

int fiveCount = 0;

for (int i = 5; i <= n; i++)

{

int tem = i;

while (tem % 5 == 0)

{

tem /= 5;

fiveCount++;

}

}

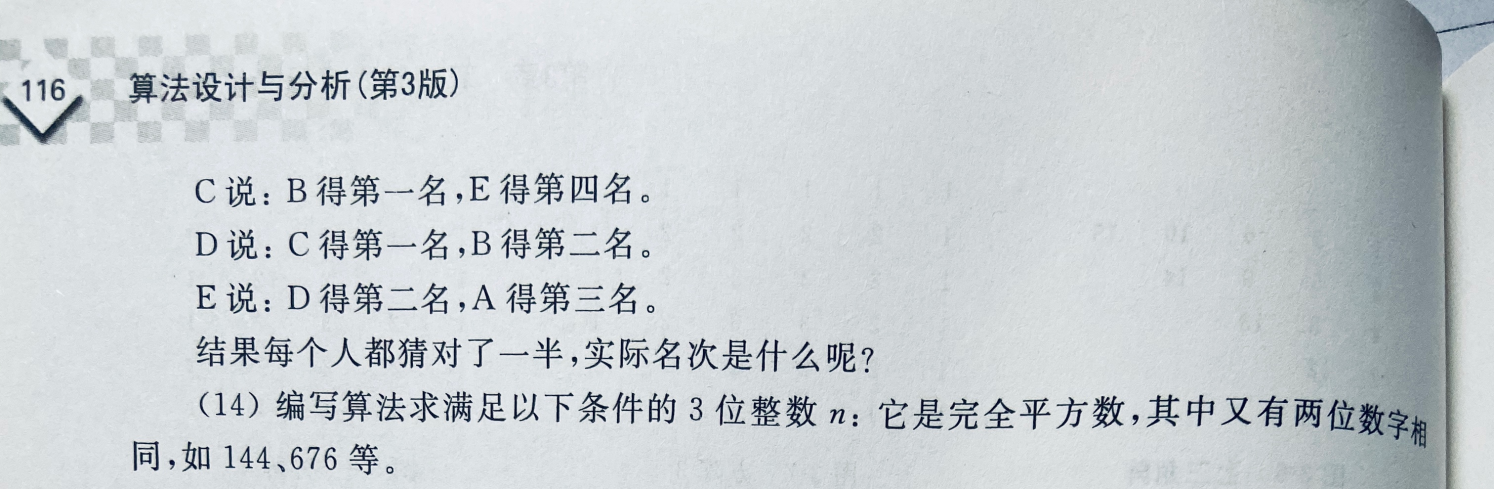
printf("末尾有%d个0",fiveCount);

}

int main() {

findZero();

}

  
13.思路：使用暴力穷举法，将六句话传化为表达式

伪代码：find mc( )

{ for a=1 to 5 do

for b=1 to 5 do

if a!=b then

for c=1 to 5 do

if c!=a & c!=b then

for d=1 to 5 do

if d!=a & d!=b & d!=c then{

e=15-a-b-c-d

if e!=a & e!=b & e!=c & e!=d

and (b=3)+(c=5)=1 and (d=2)+(e=4)=1 and (b=1)+(e=1)=1 and (c=1)+(b=2)=1 and (d=2)+(a=3)=1

Then printf(a,b,c,d,e); break

}

}

源程序：#include <stdio.h>

int main( ){

int a, b, c, d, e;

for (a = 1; a <= 5; a = a + 1)

for (b = 1; b <= 5; b = b + 1)

if (a != b)

for (c = 1; c <= 5; c = c + 1)

if (c != a && c != b)

for (d = 1; d <= 5; d = d + 1)

if (d != a && d != b && d != c)

{

e = 15 - a - b - c - d;

if (e != a && e != b && e != c && e != d)

if ((((b == 3) + (c == 5)) == 1) && (((d == 2) + (e == 4)) == 1) && (((b == 1) + (e == 4)) == 1) && (((c == 1) + (b == 2)) == 1) && (((d == 2) + (a == 3)) == 1))

printf("实际名次是：a=%d，b=%d，c=%d，d=%d，e=%d", a, b, c, d, e);

}

return 0;

}

14.思路：将为三位数的平方数全部找到，将三位数字一一比较

伪代码：int main()

for i=10 to 31 do{

num = i \* i;

tem = num;

n1 = num % 10;//个位

num /= 10;

n2 = num % 10;//十位

num /= 10;

n3 = num;//百位

if (n1 == n2 || n1 == n3 || n2 == n3) then print(tem);}

}

源代码： #include <stdio.h>

int main() {

printf("满足条件的三位整数数为：");

for (int i = 10; i <= 30; i++)

{

int num = i \* i;

int tem = num, n1, n2, n3;

n1 = num % 10;

num /= 10;

n2 = num % 10;

num /= 10;

n3 = num;

if (n1 == n2 || n1 == n3 || n2 == n3)

printf(" %d ", tem);

}

return 0;

}