**习题一**

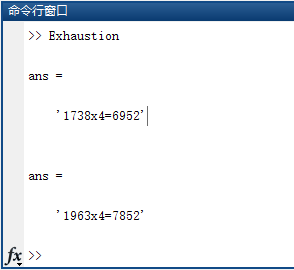
1. 解题分析

由题意，简单分析可知，第二个数属于2-9，1不可能成立。又第一个数必须在1234-4987之前，因为前者为不重复最小四位数，后者为可能成非零个位数仍为四位数的最大四维不重复数。因此，我们可以在第个位数范围内遍历（1234-4978）的基础上遍历第二个数（2-9），然后算出第三个数。然后分析三个数是否满足有且仅有一个1-9这9个数。

1. 解题过程

用for循环i 1234-4987 代表第一个数，在循环内 j 2-9 代表第二个数。c=i\*j 代表第三个数，首先由四位数可知c<10000，由spriintf函数将ijc，连为一字符串str。然后先用strfind函数依次搜索str函数中0的存在位置，由length判断存在个数（=0），然后在while函数循环中用strfind函数依次搜索str函数中1-9的存在位置，由length判断存在个数（=1）。若满足题意思，则记录在result矩阵中并显示出来。

1. 结果截图



1. 源代码

见Exhaustion.m

clear

r=1;

for i=1234:4987

for j=2:9

flag1=0;

k=1;

c=i\*j;

if c<10000

str=sprintf('%d',i,j,c);

flag1=1;

end

%%判断0不存在

t=strfind(str,'0');

if length(t)~=0

flag1=0;

end

%%判断1-9存在且仅存在一次

while k<10&flag1==1

aim=sprintf('%d',k);

t=strfind(str,aim);

if length(t)~=1

flag1=0;

end

k=k+1;

end

%%结果存储与输出

if flag1==1

result(r,:)=[i,j,c];

r=r+1;

sprintf('%dx%d=%d',i,j,c)

end

end

end