

# 数独游戏

姓名： 王继康 16029110064  
马加羽 17029110076

## 一、程序思路

数独游戏规则，填入的数字，不能与同一行数字重复，不能与同一列数字重复，不能在所属大方格中的数字重复。所以只需要判断输入数字，分别在行，列，和大方格中是否有重复数字。

## 二、算法

设计题目，由标准输入坐标和数字。分别判断坐标所在行，所在列，所在大方格是否有重复数字，如果有重复数字，输出错误信息“`placemen invalide`”，如果没有重复数字，则符合要求，将数字填入数独矩阵。

## 三、程序分析

变量

<code>jeuDeSudoku[9][9]</code>	数独游戏矩阵;
<code>x, y</code>	输入数字坐标;
<code>chiffre</code>	输入的数字;
<code>rangee</code>	所在大方格初始行
<code>colonne</code>	所在大方格初始列

函数分析

### 1. 主函数 `int main()`

主函数中定义数独游戏矩阵，未填的空白方格初始化为 0，输出题目。  
`while` 循环在 `int fin(int jeuDeSudoku)` 函数判断游戏是否结束，如果未结束进入循环。玩家输入坐标和答案，进入 `void valide(int jeuDeSudoku[][9],int x,int y,int chiffre)` 函数，判断输入答案是否符合要求，再用 `void printSudoku(int jeuDeSudoku[][9])` 函数输出数独矩阵。直到全部填满，循环结束。  
输出游戏成功，程序结束。

### 2. 判断游戏是否结束函数 `int fin(int jeuDeSudoku[][9])`

对整个数组进行遍历，如果全部数组中有 0，则证明游戏未结束，`return` 返回 0，若数组全部为 0，游戏结束，返回 1。

### 3. 输出数独矩阵函数 `void printSudoku(int jeuDeSudoku[][9])`

以 `%2d` 的形式输出数独数组，若数组值为 0，输出“.”，每行结束换行形成矩阵。

### 4. 判断有效值函数 `void valide(int jeuDeSudoku[][9],int x,int y,int chiffre)`

第一个 if, 判断输入的数字 `chiffre` 是否在 1-9 之间, 若否, 输出错误。

第二个 if, 判断所填 `x,y` 坐标上是否已被填上数字, 若已有数字, 输出错误信息。

第一个 for 循环, 判断 `chiffre` 所在列是否与 `chiffre` 重复, 若重复, 输出错误信息, `return`。

第二个 for 循环, 判断 `chiffre` 所在行是否与 `chiffre` 重复, 若重复, 输出错误信息并 `return`。

`rangee` 与 `colonne` 是 `chiffre` 所在大方格初始坐标, 通过对 `x` 和 `y` 分别对 3 整除再乘 3 可得。

第三个 for 循环, 遍历 `chiffre` 所在大方格数字, 判断是否与 `chiffre` 重复, 若重复, 输出错误信息并 `return`。

若以上判断都没问题, 将 `chiffre` 填入数独矩阵。