单偶数阶幻方生成器作业完成报告

> 学号：1651162
>
> 姓名：施程航
>
> 语言：c++
>
> 开发环境：windows visual studio

- 需求分析

  输入1-4中的某个数字x，输出显示4x+2阶的幻方结果。(即所有**20**以内奇数阶的幻方)

- 解题思路

  单偶阶幻方是属于较复杂的一种幻方，本项目采用**斯特雷奇(Ralph Strachey)单偶阶幻方构造法**。单偶阶指的是所求幻方阶数n为偶数，且不能被4整除(n = 6,10,14,18)。取n=m*4+2 h=n/2(h=2*m+1)，该构造方法如下： 1.把方阵分为A,B,C,D四个部分，每个方阵的行和列的大小均为h，即各个方阵均为一个奇数阶方阵，记左上、右上、左下、右下四个方阵为A B C D。在A方阵中把1 -- h*h按照奇数阶幻方的方法填入，B C D方阵则将A方阵上对应的位置上的数再分别加上2*h*h、h*h、3*h*h。以n=10为例，执行完此步骤后得到方阵如下：

```
17   24    1    8   15   67   74   51   58   65
23    5    7   14   16   73   55   57   64   66
4     6   13   20   22   54   56   63   70   72
10   12   19   21    3   60   62   69   71   53
11   18   25    2    9   61   68   75   52   59
92   99   76   83   90   42   49   26   33   40
98   80   82   89   91   48   30   32   39   41
79   81   88   95   97   29   31   38   45   47
85   87   94   96   78   35   37   44   46   28
86   93  100   77   84   36   43   50   27   34
```

2.A象限的中间行、中间格(也即在大方阵的m行m列)，按从左到右的方向选出m格。A象限的其他行则标出最左边的m格，所有标记出来的数和C象限对应位置上的数交换位置。以n=10为例，执行完此步骤后得到方阵如下：

```
92   99    1    8   15   67   74   51   58   65
98   80    7   14   16   73   55   57   64   66
4     6   88   95   22   54   56   63   70   72
85   87   19   21    3   60   62   69   71   53
86   93   25    2    9   61   68   75   52   59
17   24   76   83   90   42   49   26   33   40
23    5   82   89   91   48   30   32   39   41
79   81   13   20   97   29   31   38   45   47
10   12   94   96   78   35   37   44   46   28
11   18  100   77   84   36   43   50   27   34
```

3.在B象限的所有行的中间格子，从左到右，标出m-1(比如10阶方阵则是一格)个格子，所有标出的格子和D象限相应位置上的数进行交换，幻方形成，以n=10为例，执行完此步骤后得到方阵如下：

```
92   99    1    8   15   67   74   26   58   65
98   80    7   14   16   73   55   32   64   66
4    6   88   95   22   54   56   38   70   72
85   87   19   21    3   60   62   44   71   53
86   93   25    2    9   61   68   50   52   59
17   24   76   83   90   42   49   51   33   40
23    5   82   89   91   48   30   57   39   41
79   81   13   20   97   29   31   63   45   47
10   12   94   96   78   35   37   69   46   28
11   18  100   77   84   36   43   75   27   34
```

- 算法设计
  - 生成单阶幻方

    单阶幻方是斯特雷奇构造法中构造偶数阶幻方的基础，单阶幻方的构造方法较为简单：

    - 1.将1放在第一行中间的格子上
    - 2.从2 -- n*n为止，按45°方向比如向右上行走(也即每一个数存放的行比签一个属的行数减一，列数加一)
    - 3.如果行数或列数超出方阵范围，则回绕(也即-1变为n-1或者n变为0，默认下表从0开始)
    - 4.如果按2、3规则找到的位置上已经有数，则把当前的数放在该位置的下面

```
void process_odd(int n, std::vector<std::vector<int > > &container)
{
    assert(n % 2 == 1);
    //std::vector<std::vector<int> > container(n, std::vector<int>(n,
0));

    int row = 0, col = n / 2;
    for (int num = 1;num <= n * n;++num) {
        container[row][col] = num;
        if (container[(row + n - 1) % n][(col + 1) % n] == 0) {
            row = (row + n - 1) % n;
            col = (col + 1) % n;
        }
        else
            row = (row + 1) % n;
    }

    //display(container);
}
```

  - 运行算法 按照斯特雷奇法的步骤进行填数和换数

```
std::vector<std::vector<int> > process_single_even(int n)
{
    assert(n % 4 == 2 && n != 2);

    std::vector<std::vector<int> > container(n, std::vector<int>(n,
0));

    int half_n = n / 2;
    //step1
    process_odd(half_n, container);
    for (int row = 0;row < half_n;++row)
        for (int col = 0;col < half_n;++col) {
            int cur = container[row][col];
            container[row][col + half_n] = cur + 2 * half_n*half_n;
            container[row + half_n][col] = cur + 3 * half_n*half_n;
            container[row + half_n][col + half_n] = cur + half_n *
half_n;
        }
    //display(container);
    //5 2
    using std::swap;
    int quarter = (n - 2) / 4;
    //step2
    for (int row = 0;row < half_n;++row)
        for (int col = 0;col < quarter;++col) {
            if (row != quarter)
                swap(container[row][col], container[row + half_n]
[col]);
            else
                swap(container[row][col + quarter],
                    container[row + half_n][col + quarter]);
    }
    //display(container);
    //step3
    for (int row = 0;row < half_n;++row) {
        for (int col = n - quarter - 1;col > half_n + 1;--col)
            swap(container[row][col], container[row + half_n][col]);
    }

    check(container);

    std::cout << "\n"<< n <<"阶幻方: :\n";
    display(container);
    std::cout << std::string(50, '=') << "\n\n";

    return container;
}
```

- **检查幻方是否正确生成** 检查分为四个步骤进行：
  - 1.检查方阵各行的和是否相等

- 2.检查方阵各列的和是否相等
- 3.检查方阵的两条对角阵的和是否相等
- 4.检查列的和、行的和以及对角线的和是否均相等

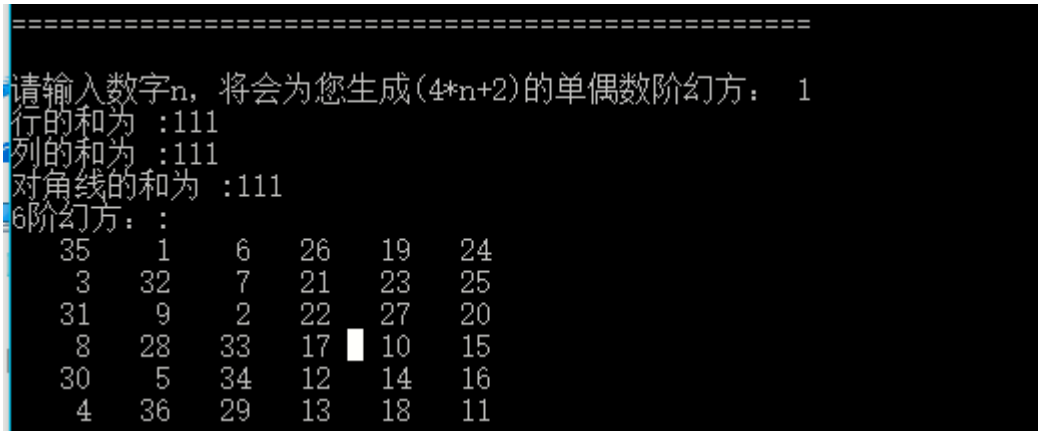
若条件1 2 3 4均满足那么生成的方阵是正确的

```
bool check(std::vector<std::vector<int> >& container)
{
    int row_size = container.size();
    int col_size = container[0].size();//actually, the row_size and
col_size should be equal
    //
    int row_sum = 0,
    col_sum = 0,
    diagonal_sum_1 = 0, diagonal_sum_2 = 0;

    //calculate row_sum
    for (int row = 0;row < row_size;++row) {
        int temp_sum = 0;
        for (int col = 0;col < col_size;++col)
            temp_sum += container[row][col];
        if (row_sum != 0 && row_sum != temp_sum) {
            std::cout << "各行的和不相等\n";
            return false;
        }
        //update row_sum
        row_sum = temp_sum;
    }
    //calculate col_sum
    for (int col = 0;col < col_size;++col) {
        int temp_sum = 0;
        for (int row = 0;row < row_size;++row)
            temp_sum += container[row][col];
        if (row_sum != 0 && row_sum != temp_sum) {
            std::cout << "各列的和不相等\n";
            return false;
        }
        //update
        col_sum = temp_sum;
    }
    //diagonal_sum
    for (int step = 0;step < row_size;++step) {
        diagonal_sum_1 += container[step][step];
        diagonal_sum_2 += container[step][col_size - step - 1];
    }
    if (diagonal_sum_1 != diagonal_sum_2) {
        std::cout << "两对角线的和不相等!\n";
        return false;
    }
    if (row_sum != col_sum || row_sum != diagonal_sum_1) {
        std::cout << "行、列、对角线的值不全相等!\n";
```

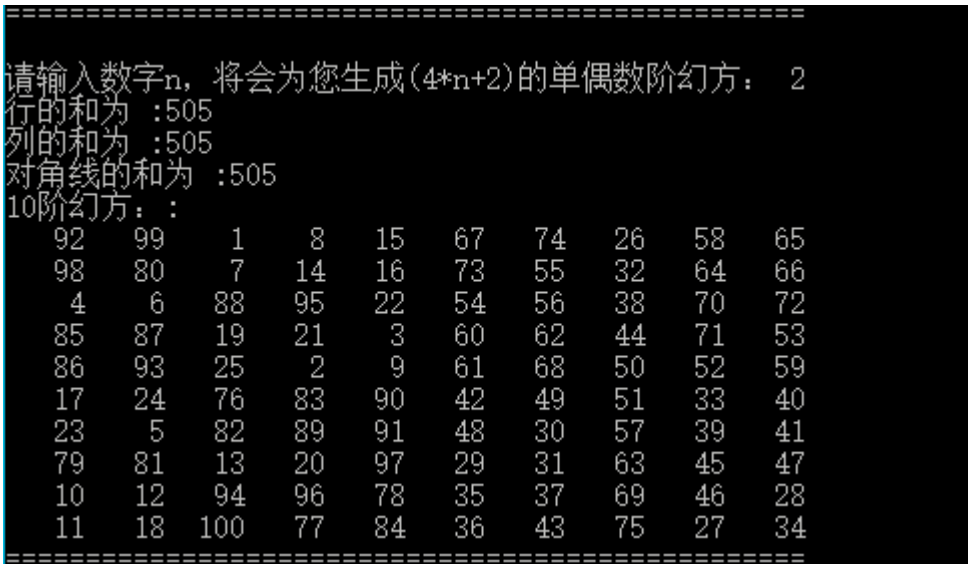
```
            return false;
        }
        std::cout << "行的和为 :" << row_sum;
        std::cout << "\n列的和为 :" << col_sum;
        std::cout << "\n对角线的和为 :" << diagonal_sum_1;

        return true;
    }
```

- 结果分析
    - 用程序接受所要求的范围内输出均正确

1.n=1，生成6阶方阵



2.n=2，生成10阶方阵



3.n=3，生成14阶方阵

```
==================================================

请输入数字n，将会为您生成(4*n+2)的单偶数阶幻方： 3
行的和为 :1379
列的和为 :1379
对角线的和为 :1379
14阶幻方：：
 177  186  195    1   10   19   28  128  137   97   50  108  117  126
 185  194  154    9   18   27   29  136  145   56   58  116  125  127
 193  153  155   17   26   35   37  144  104   57   66  124  133  135
   5   14   16  172  181  183   45  103  112   65   74  132  134  143
 160  162  171   33   42   44    4  111  113   73   82  140  142  102
 168  170  179   41   43    3   12  119  121   81   90  141  101  110
 169  178  187   49    2   11   20  120  129   89   98  100  109  118
  30   39   48  148  157  166  175   79   88  146   99   59   68   77
  38   47    7  156  165  174  176   87   96  105  107   67   76   78
  46    6    8  164  173  182  184   95   55  106  115   75   84   86
 152  161  163   25   34   36  192   54   63  114  123   83   85   94
  13   15   24  180  189  191  151   62   64  122  131   91   93   53
  21   23   32  188  190  150  159   70   72  130  139   92   52   61
  22   31   40  196  149  158  167   71   80  138  147   51   60   69
==================================================
```

4.n=4，生成18阶方阵

```
==================================================

请输入数字n，将会为您生成(4*n+2)的单偶数阶幻方： 4
行的和为 :2925
列的和为 :2925
对角线的和为 :2925
18阶幻方：：
 290  301  312  323    1   12   23   34   45  209  220  150  161   82  174  185  196  207
 300  311  322  252   11   22   33   44   46  219  230  160   90   92  184  195  206  208
 310  321  251  253   21   32   43   54   56  229  240   89   91  102  194  205  216  218
 320  250  261  263   31   42   53   55   66  239  169   99  101  112  204  215  217  228
   6   17   19   30  284  295  306  308   76  168  179  100  111  122  214  225  227  238
 259  270  272  283   51   62   64   75    5  178  189  110  121  132  224  226  237  167
 269  271  282  293   61   72   74    4   15  188  190  120  131  142  234  236  166  177
 279  281  292  303   71   73    3   14   25  198  200  130  141  152  235  165  176  187
 280  291  302  313   81    2   13   24   35  199  210  140  151  162  164  175  186  197
  47   58   69   80  244  255  266  277  288  128  139  231  242  163   93  104  115  126
  57   68   79    9  254  265  276  287  289  138  149  241  171  173  103  114  125  127
  67   78    8   10  264  275  286  297  299  148  159  170  172  183  113  124  135  137
  77    7   18   20  274  285  296  298  309  158   88  180  182  193  123  134  136  147
 249  260  262  273   41   52   63   65  319   87   98  181  192  203  133  144  146  157
  16   27   29   40  294  305  307  318  248   97  108  191  202  213  143  145  156   86
  26   28   39   50  304  315  317  247  258  107  109  201  212  223  153  155   85   96
  36   38   49   60  314  316  246  257  268  117  119  211  222  233  154   84   95  106
  37   48   59   70  324  245  256  267  278  118  129  221  232  243   83   94  105  116
==================================================
```

- 对于n=5程序也能很好工作

5.n=5，生成22阶方阵

```
------------------------------------------------

请输入数字n，将会为您生成(4*n+2)的单偶数阶幻方： 5
行的和为 :5335
列的和为 :5335
对角线的和为 :5335
22阶幻方：:
 431  444  457  470  483    1   14   27   40   53   66  310  323  215  228  241  122  256  269  282  295  308
 443  456  469  482  374   13   26   39   52   65   67  322  335  227  240  132  134  268  281  294  307  309
 455  468  481  373  375   25   38   51   64   77   79  334  347  239  131  133  146  280  293  306  319  321
 467  480  372  385  387   37   50   63   76   78   91  346  359  130  143  145  158  292  305  318  320  333
 479  371  384  386  399   49   62   75   88   90  103  358  250  142  144  157  170  304  317  330  332  345
   7   20   33   35   48  424  437  450  452  465  115  249  262  154  156  169  182  316  329  331  344  357
 382  395  397  410  423   73   86   99  101  114    6  261  274  155  168  181  194  328  341  343  356  248
 394  407  409  422  435   85   98  100  113    5   18  273  286  167  180  193  206  340  342  355  247  260
 406  408  421  434  447   97  110  112    4   17   30  285  287  179  192  205  218  352  354  246  259  272
 418  420  433  446  459  109  111    3   16   29   42  297  299  191  204  217  230  353  245  258  271  284
 419  432  445  458  471  121    2   15   28   41   54  298  311  203  216  229  242  244  257  270  283  296
  68   81   94  107  120  364  377  390  403  416  429  189  202  336  349  362  243  135  148  161  174  187
  80   93  106  119   11  376  389  402  415  428  430  201  214  348  361  253  255  147  160  173  186  188
  92  105  118   10   12  388  401  414  427  440  442  213  226  360  252  254  267  159  172  185  198  200
 104  117    9   22   24  400  413  426  439  441  454  225  238  251  264  266  279  171  184  197  199  212
 116    8   21   23   36  412  425  438  451  453  466  237  129  263  265  278  291  183  196  209  211  224
 370  383  396  398  411   61   74   87   89  102  478  128  141  275  277  290  303  195  208  210  223  236
  19   32   34   47   60  436  449  462  464  477  369  140  153  276  289  302  315  207  220  222  235  127
  31   44   46   59   72  448  461  463  476  368  381  152  165  288  301  314  327  219  221  234  126  139
  43   45   58   71   84  460  473  475  367  380  393  164  166  300  313  326  339  231  233  125  138  151
  55   57   70   83   96  472  474  366  379  392  405  176  178  312  325  338  351  232  124  137  150  163
  56   69   82   95  108  484  365  378  391  404  417  177  190  324  337  350  363  123  136  149  162  175
================================================
```

- 理论上程序能生成任意阶奇偶阶方阵

- 结论
  - 通过斯特雷奇法可以正确生成奇偶阶幻方
  - 在该构造法中，奇偶阶幻方的工作正确性基于奇数阶幻方的正确性，说明不同类型的幻方之间存在着一定的联系
  - 生成幻方后需要对幻方进行正确性的检查，以保证算法没有错误

- 参考

  奇数阶、偶数阶幻方制作方法