

WPC French  
Qualifier 2010

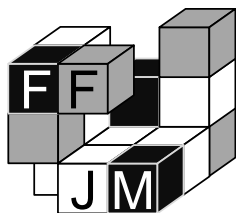
**Part II**

Name

1. Easy as ABC
2. All Alone
3. Minesweeper
4. Fences
5. Tents
6. Meanders
7. Kropki
8. Figure Cut
9. Subset Sum
10. Loop Finder

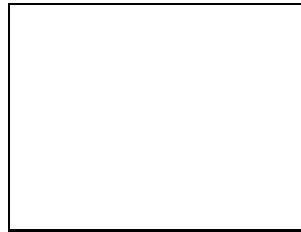
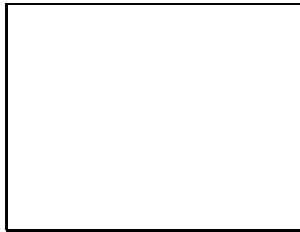
|                       |          |          |           |           |           |           |           |           |           |           |           |
|-----------------------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <b>Puzzles solved</b> | <b>0</b> | <b>1</b> | <b>2</b>  | <b>3</b>  | <b>4</b>  | <b>5</b>  | <b>6</b>  | <b>7</b>  | <b>8</b>  | <b>9</b>  | <b>10</b> |
| <b>Points</b>         | <b>0</b> | <b>5</b> | <b>10</b> | <b>16</b> | <b>23</b> | <b>30</b> | <b>38</b> | <b>46</b> | <b>55</b> | <b>65</b> | <b>75</b> |

**Total: 75 points + time bonus (5 pts/minute)**



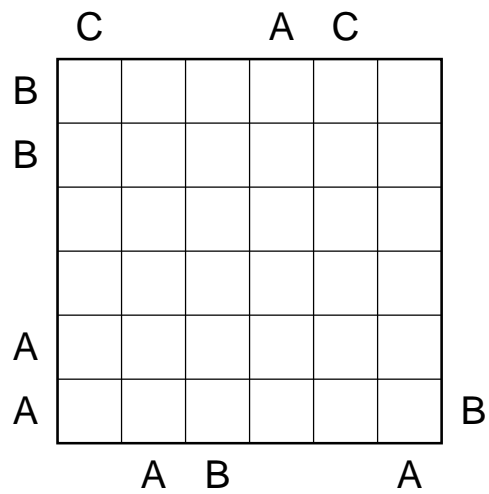
WPC French  
Qualifier 2010

**Part II**



**1. Easy as ABC**

Place the letters A, B, C into the diagram, so that each letter occurs once in each row and column. The letters outside the diagram indicate the first letter seen from that direction.



**WPC French  
Qualifier 2010**

**Part II**

**2. All Alone**

Black out some of the numbers in the grid so that each row and each column contains only different digits. Black squares must not touch horizontally or vertically, and the remaining squares must all be connected to each other.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 4 | 3 | 5 | 3 | 4 | 2 | 7 |
| 5 | 2 | 6 | 6 | 1 | 4 | 2 |
| 3 | 5 | 4 | 2 | 3 | 4 | 4 |
| 3 | 4 | 2 | 5 | 2 | 1 | 6 |
| 7 | 2 | 3 | 1 | 4 | 4 | 2 |
| 2 | 3 | 7 | 1 | 5 | 6 | 1 |
| 1 | 5 | 4 | 6 | 2 | 6 | 5 |

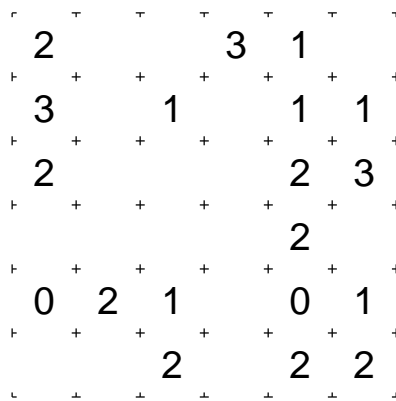


WPC French  
Qualifier 2010

**Part II**

**4. Fences**

Draw a single continuous loop by connecting neighboring dots horizontally or vertically (but not diagonally). A numbered square indicates exactly how many of its four edges are used by the loop.



WPC French  
Qualifier 2010

Part II

5. Tents

Locate the tents in the grid. Each tree is connected to exactly one tent, found in a horizontally or vertically adjacent square. Tents do not touch each other, not even diagonally. The numbers outside the grid reveal the total number of tents in the corresponding row or column.

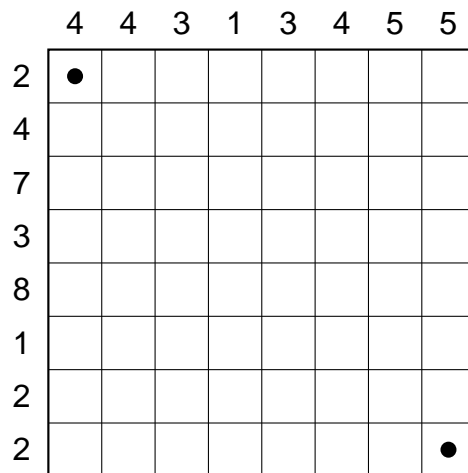
|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | 2 | 1 | 2 | 1 | 3 | 1 | 2 |
| 2 |   | 🌳 |   |   |   | 🌳 |   |
| 2 |   |   |   |   |   | 🌳 |   |
| 2 | 🌳 |   |   | 🌳 |   |   |   |
| 1 |   | 🌳 |   |   | 🌳 | 🌳 |   |
| 2 |   |   | 🌳 |   |   |   |   |
| 1 |   |   | 🌳 |   |   |   | 🌳 |
| 2 |   |   |   | 🌳 |   |   |   |

WPC French  
Qualifier 2010

**Part II**

**6. Meanders**

Find a route starting in the top left square and ending in the bottom right square. The route meanders horizontally or vertically, and the numbers outside the grid indicate the number of occupied squares in that row or column.

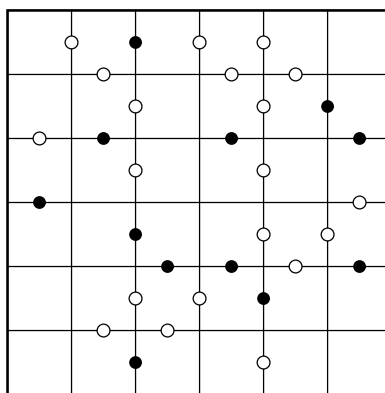


WPC French  
Qualifier 2010

**Part II**

**7. Kropki**

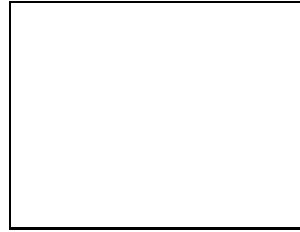
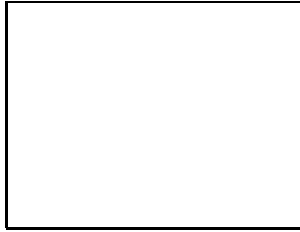
Fill the table with digits from 1 to 6, so that each digit appears in every row and column exactly once. If the absolute difference between the digits in two adjacent cells equals 1 then they are separated by a white dot. If the digits in two adjacent cells are exactly half of each other then they are separated by a black dot. The dot between adjacent cells containing “1” and “2” can be either white or black.





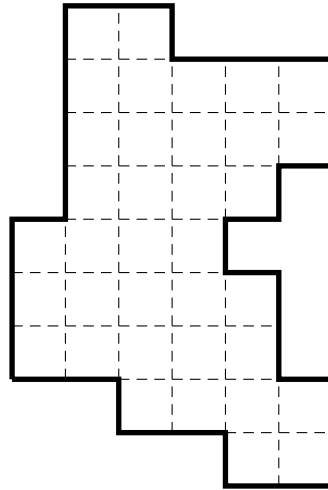
**WPC French  
Qualifier 2010**

**Part II**



### **8. Figure Cut**

Cut the figure into 6 identical pieces. The pieces may be rotated, but cannot be reflected.



**WPC French  
Qualifier 2010**

**Part II**

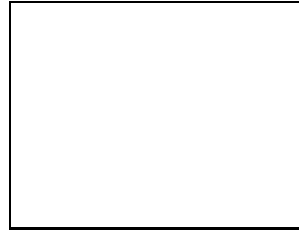
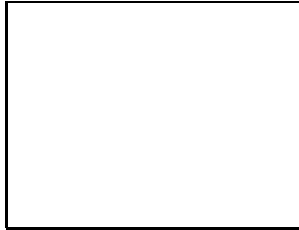
### **9. Subset Sum**

Find a subset of the given numbers which add up to exactly 100.

12 15 22 29 33 36 40 43

**WPC French  
Qualifier 2010**

**Part II**



### **10. Loop Finder**

Draw a continuous loop formed by straight line segments connecting the centers of adjacent squares. The loop must not cross or overlap itself, and must visit all squares. Some parts of the loop are already given.

