Paths on a grid

[Year 2016-2017]

by GALL Ștefania, GRUMĂZESCU George Octavian, LEFTER Lucia-Maria, OANȚĂ Ana Estera, STOICA George-Ioan, ZAHARIA Medeea Maria, 7th grade
from Colegiul Național „Costache Negruzzi" Iași

Teacher: Adrian ZANOSCHI
Researcher: Iulian STOLERIU, Faculty of Mathematics,
„Alexandru Ioan Cuza" University of Iași

Keywords: path, grid, counting, product rule, the inclusion-exclusion principle.

1. PRESENTATION OF THE RESEARCH TOPIC

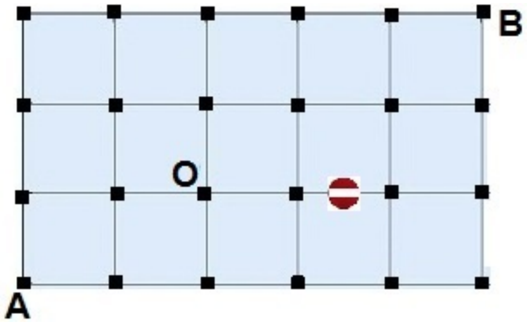
In this article, we are counting paths that join two opposite corners of a rectangular grid. The grid is supposed to represent the street map of a town center. Two grid sizes and various traffic restrictions are considered.

2. BRIEF PRESENTATION OF THE CONJECTURES AND RESULTS OBTAINED

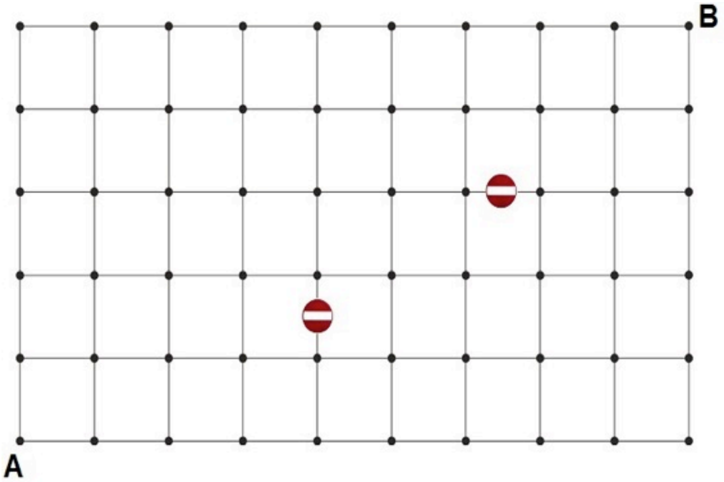
In the following, we will agree that:

node = the intersection point of two lines of the map;
street = a segment that unites two adjacent nodes of the map;
path = a succession of streets.

Diagrams (α) and (β) below represent the mapping of the streets from the center of two towns, *Little Rock* and, respectively, *Big Hill*. Each side of a square represents a new street, and in each node one touristic attraction exists. On each of these streets, you can walk (as a pedestrian) in any direction or you can drive only in two directions, rightward or upward.



(α) : The *Little Rock* town center



(β): The *Big Hill* town center

We intend to address the following problems:

(a) In how many ways can one drive from point A to point B of the *Little Rock* town without counting the possible restrictions from the route?

(a_1) Same question as in (a), but with the restriction to pass through attraction O.

(a_2) Same question as in (a), taking into account that the traffic is closed on the street marked in Figure (α) (in other words, the respective square becomes inaccessible).

(a_3) For which street, supposing it is closed, the number of ways of driving from A to B is minimal? Is this the only street having this property?

On two of the *Big Hill* town streets (marked in the figure) access is forbidden for cars and pedestrians alike.

(b) In how many ways can one drive from point A to point B of *Big Hill*, taking into account the restrictions?

(c) Two walkers, *Alin* and *Bianca*, depart simultaneously from points A and B, respectively. *Alin* may move only one square right or upwards, while *Bianca* may move only one square left or downwards (when looking at the grid from the same perspective). For each of them, the choice of direction is made with equal probabilities in every intersection. What are the odds that *Alin* and *Bianca* meet?

(d) Apart from the two above mentioned restrictions, other streets in the town will be under repair and will be inaccessible for a while. Find out the maximum number of streets that can be under repair simultaneously, so that all the touristic objectives are connected to point A by at least one path. We maintain the original rules of movement. The path starting at A may end in any point (touristic objective).

We shall solve the traffic problems in two ways: analytically, by using tools from combinatorics and numerically, using C++ programming.

3. THEORETICAL NOTIONS

A. Let $n \in \bullet^{*}$. The product $n! = 1 \cdot 2 \cdot ... \cdot n$ is called *n factorial*. By convention, $0! = 1$.

We observe that $n! = (n-1)! \cdot n$, for any $n \in \bullet^{*}$.

Let $n \in \bullet^{*}$, $k \in \bullet, k \leq n$. The number of subsets which have k elements of a set with n elements is named *combinations of n elements taken k at a time* and it is written as $\binom{n}{k}$.

Of course, $\binom{n}{k} = \binom{n}{n-k}$, because choosing a subset with k elements is equivalent to choosing a subset with $n-k$ elements to be left out.

It is known that $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$, for any $n \in \bullet^{*}$ and any $k \in \bullet, k \leq n$.

B. Theorem (*Principle of inclusion-exclusion for two sets*) If A and B are two finite sets, then

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

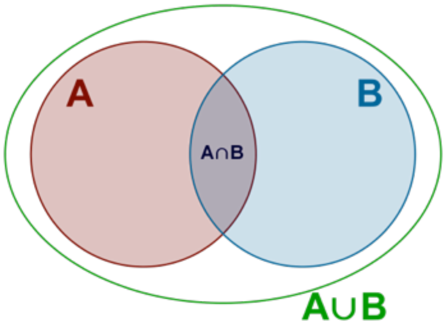


Figure 1

In other words, given 2 finite sets, the number of elements belonging to at least one of the sets is equal to the sum of the number of elements in each set, from which we subtract the number of elements common to the 2 sets.

C. *The multiplication principle (the product rule)*. If A and B are two sets and $A\times B$ notes their Cartesian product (i.e. $A\times B=\{(a,b)\mid a\in A, b\in B\}$), then

$$|A\times B|=|A|\cdot|B|. \tag{1.1}$$

Otherwise stated, given two finite sets, the number of ways in which one can form ordered pairs in which the first element is in the first set and the second element is in the second set is equal to the product of the number of elements of the two sets.

Formula (1.1) generalizes to a finite number of sets $A_1, A_2, \mathrm{K}\ , A_n$:

$$|A_1\times A_2\times \mathrm{K}\ \times A_n|=|A_1|\cdot|A_2|\cdot \mathrm{K}\ \cdot|A_n|.$$

4. THE SOLUTION

(a) In how many ways can one drive from point *A* to point *B* of the *Little Rock* town without counting the possible restrictions from the route?

First method

We are computing the number of paths starting from *A* and ending in *B*, satisfying the requirements in the problem. Figure 2 illustrates how we must proceed:

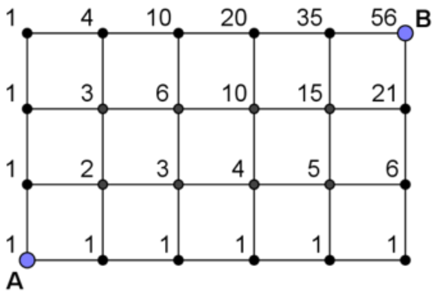


Figure 2

So we observe that the number of paths from *A* to *B* which respect all the conditions is *56*.

Second method

Because a car can move across streets only in two directions, rightward or upward, to arrive from point *A* to point *B* we must go five streets to the right and three streets upward, in a certain order. To describe the path, we can picture that we have five R letters and three U letters. When the car goes to the right we write R and when it goes upward we write U. For each possible route we obtain a unique sequence of eight letters. Figure 3 demonstrates this.

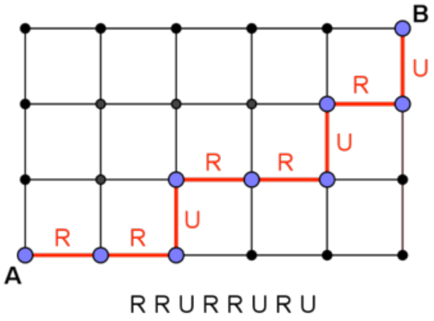


Figure 3

To calculate the number of possible route returns we calculate the number of different sequences which are written with five R letters and three U letters.

We observe that to construct a sequence knowing the position of the three U letters is enough. So it is enough to know which are the 3 numbers from 1 to 8 which correspond to the position of the U letters in the sequence. These 3 numbers form a subset with 3 elements from the set $P = \{1,2,3,4,5,6,7,8\}$. In conclusion, the number of possible routes is the number of subsets with 3 elements from the set P with 8 elements, that is $\binom{8}{3} = \frac{8!}{3! \cdot 5!} = \frac{6 \cdot 7 \cdot 8}{1 \cdot 2 \cdot 3} = 56$.

More general, if the map was a rectangle with length composed by a number of streets and the width being a b number of streets, the number of routes from which a driver can choose to get from one corner of the map to the opposite corner would be $\binom{a+b}{a} = \binom{a+b}{b} = \frac{(a+b)!}{a! \cdot b!}$.

<u>Third method: C++ program</u>

Let us consider the grid from Figure 4:

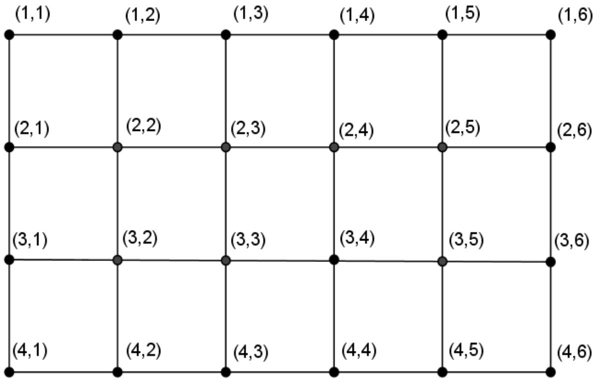
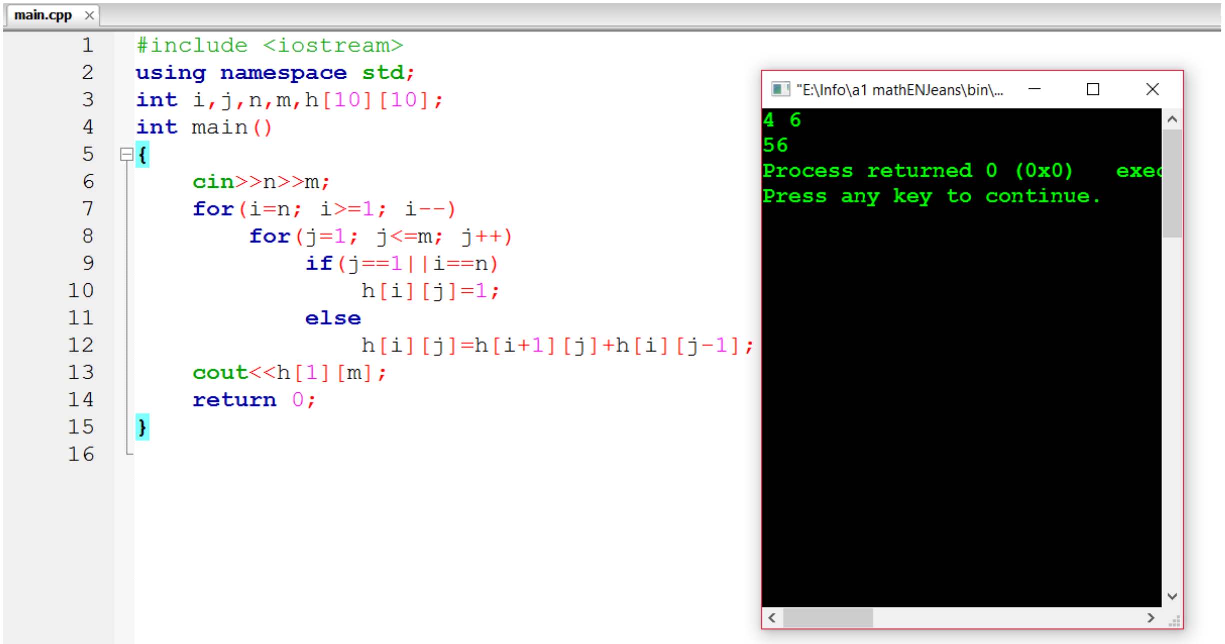


Figure 4



```cpp
#include <iostream>
using namespace std;
int i,j,n,m,h[10][10];
int main()
{
    cin>>n>>m;
    for(i=n; i>=1; i--)
        for(j=1; j<=m; j++)
            if(j==1||i==n)
                h[i][j]=1;
            else
                h[i][j]=h[i+1][j]+h[i][j-1];
    cout<<h[1][m];
    return 0;
}
```

```
4 6
56
Process returned 0 (0x0)   exec
Press any key to continue.
```

For *n*=4 and *m*=*6*, we obtain *56*. In the program *n* is the number of vertical nodes, *m* is the number of horizontal nodes and the element $h[i][j]$ of the matrix *h* stores how many paths arrive in the node (i, j), with $1 \le i \le n$ and $1 \le j \le m$ (node coordinates are given in Figure 4).

(a_1) **Same question as in (a), but with the restriction to pass through attraction O.**

<u>First method</u>

Each path from *A* to *B* is composed by a path from *A* to *O* and a path from *O* to *B* (paths which respect the conditions) (Figure 5).

The path's map from *A* to *O* is a rectangle with length composed by 2 streets and width composed by a street, so the number of possible routes is $\binom{1+2}{1} = \binom{3}{1} = 3$. The path's map from *O* to *B* is a rectangle with length composed by 3 streets and width composed by 2 streets, so the number of possible routes is $\binom{2+3}{2} = \binom{5}{2} = \frac{5!}{2! \cdot 3!} = 10$ (see Figure 6).

In conclusion, according to *the product rule*, the number of routes from *A* to *B*, which pass through *O*, is $3 \cdot 10 = 30$.

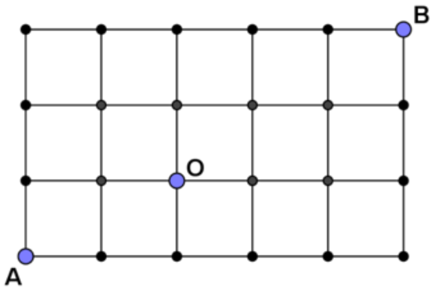


Figure 5

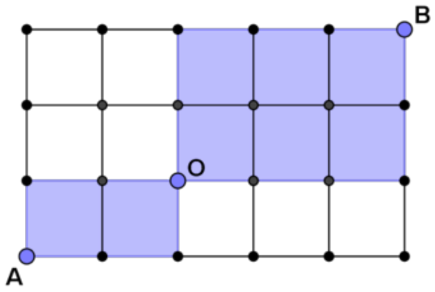
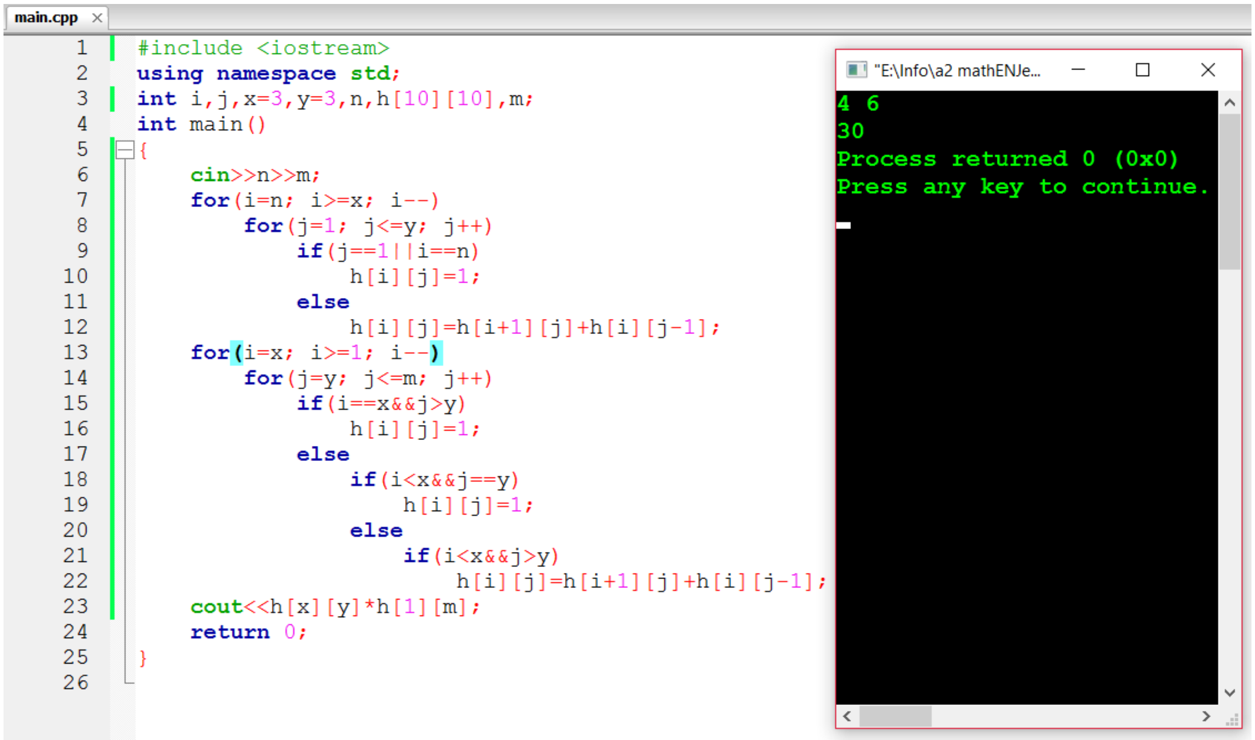


Figure 6

Second method : C++ program

In addition to the first program, (x, y) is the node we must pass through.



```
#include <iostream>
using namespace std;
int i,j,x=3,y=3,n,h[10][10],m;
int main()
{
    cin>>n>>m;
    for(i=n; i>=x; i--)
        for(j=1; j<=y; j++)
            if(j==1||i==n)
                h[i][j]=1;
            else
                h[i][j]=h[i+1][j]+h[i][j-1];
    for(i=x; i>=1; i--)
        for(j=y; j<=m; j++)
            if(i==x&&j>y)
                h[i][j]=1;
            else
                if(i<x&&j==y)
                    h[i][j]=1;
                else
                    if(i<x&&j>y)
                        h[i][j]=h[i+1][j]+h[i][j-1];
    cout<<h[x][y]*h[1][m];
    return 0;
}
```

```
4 6
30
Process returned 0 (0x0)
Press any key to continue.
```

For n=4 and m=6, we obtain *30 paths*.

(a_2) **Same question as in (a), taking into account that the traffic is closed on street *CD* (see Figure 7) (in other words, the respective square becomes inaccessible).**

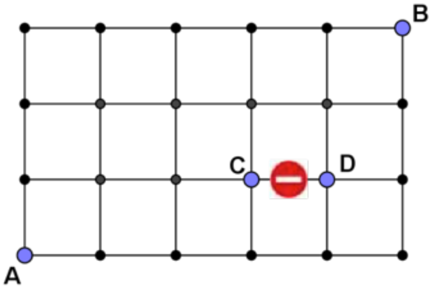


Figure 7

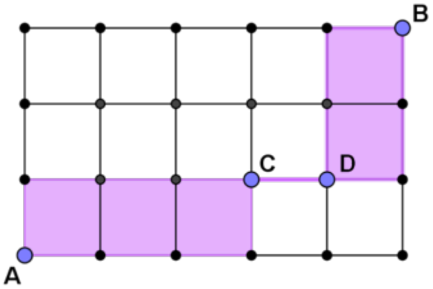


Figure 8

<u>First method</u>

One may count the number of paths that verify the restriction by subtracting the number of paths from A to B (which is *56*, according to point (a)) and the number of paths that contain street *CD* closed to traffic .

A path that contains street *CD* is composed of a path from A to C, street *CD* and a path from D to B (see Figure 8).

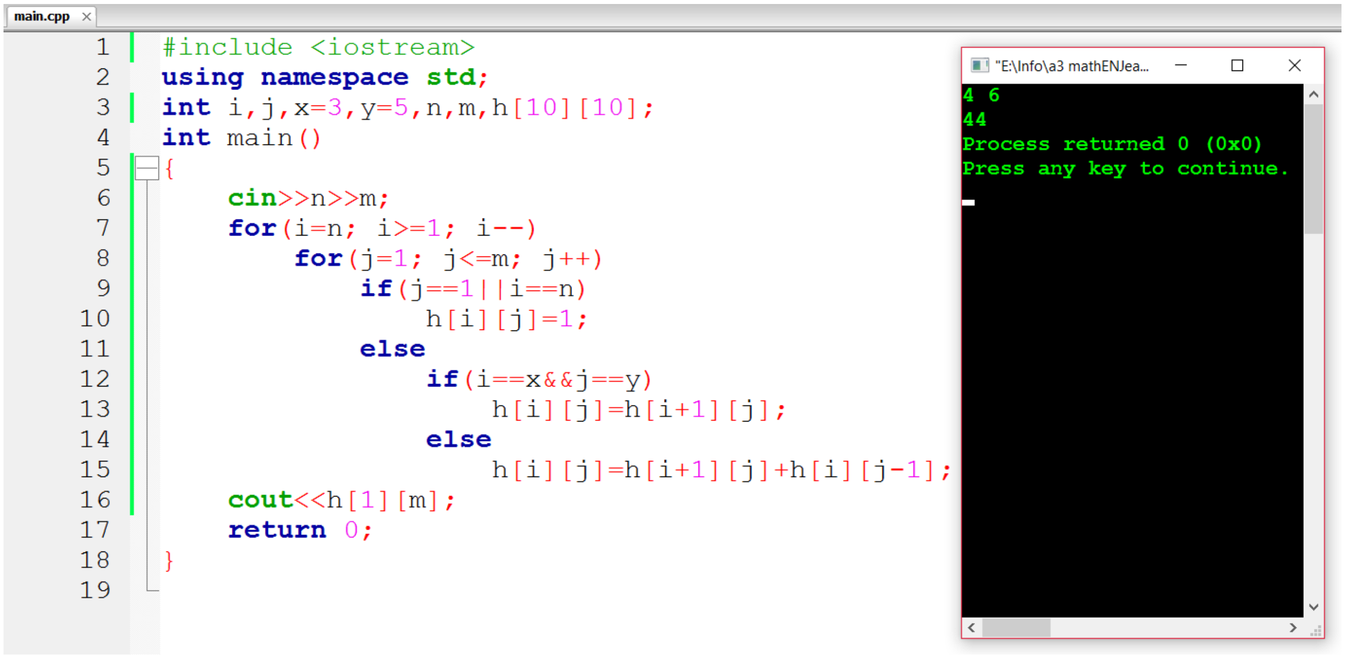
The number of paths from A to C is $\binom{3+1}{1}=\binom{4}{1}=4$. The number of paths from D to B is $\binom{2+1}{2}=\binom{3}{2}=3$.

Summing these, there are $4\cdot1\cdot3=12$ paths which contain street *CD*.

In conclusion, the number of paths which avoid street CD is $56-12=44$.

<u>Second method : C++ program</u>

In this program the coordinates of point C are (x,y) and the coordinates of point D are, of course, $(x,y+1)$.



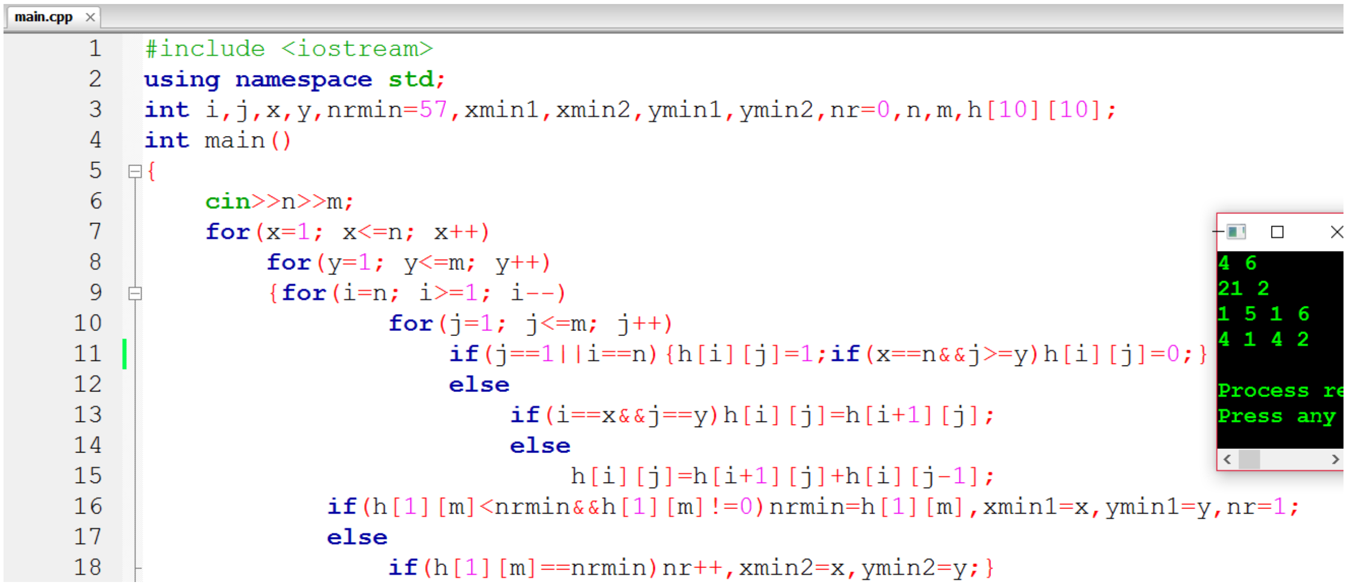
```cpp
#include <iostream>
using namespace std;
int i,j,x=3,y=5,n,m,h[10][10];
int main()
{
    cin>>n>>m;
    for(i=n; i>=1; i--)
        for(j=1; j<=m; j++)
            if(j==1||i==n)
                h[i][j]=1;
            else
                if(i==x&&j==y)
                    h[i][j]=h[i+1][j];
                else
                    h[i][j]=h[i+1][j]+h[i][j-1];
    cout<<h[1][m];
    return 0;
}

```

```
4 6
44
Process returned 0 (0x0)
Press any key to continue.
```

For n=4 and m=6, we obtain *44 paths*.

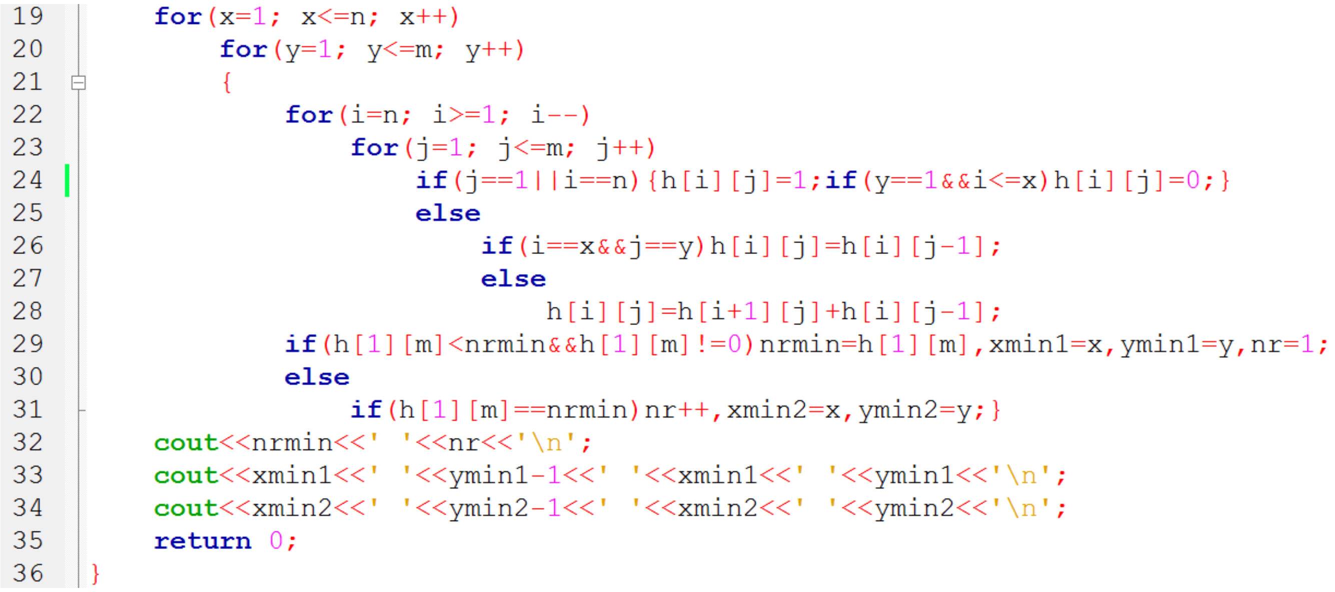
(a_3)**For which street, supposing it is closed, the number of ways of driving from *A* to *B* is minimal? Is this the only street having this property?**

<u>First method</u>* : *<u>C++ program</u>



```
#include <iostream>
using namespace std;
int i,j,x,y,nrmin=57,xmin1,xmin2,ymin1,ymin2,nr=0,n,m,h[10][10];
int main()
{
    cin>>n>>m;
    for(x=1; x<=n; x++)
        for(y=1; y<=m; y++)
        {for(i=n; i>=1; i--)
                for(j=1; j<=m; j++)
                    if(j==1||i==n){h[i][j]=1;if(x==n&&j>=y)h[i][j]=0;}
                    else
                        if(i==x&&j==y)h[i][j]=h[i+1][j];
                        else
                            h[i][j]=h[i+1][j]+h[i][j-1];
            if(h[1][m]<nrmin&&h[1][m]!=0)nrmin=h[1][m],xmin1=x,ymin1=y,nr=1;
            else
                if(h[1][m]==nrmin)nr++,xmin2=x,ymin2=y;}
```

```
4 6
21 2
1 5 1 6
4 1 4 2

Process re
Press any
```



```
    for(x=1; x<=n; x++)
        for(y=1; y<=m; y++)
        {
            for(i=n; i>=1; i--)
                for(j=1; j<=m; j++)
                    if(j==1||i==n){h[i][j]=1;if(y==1&&i<=x)h[i][j]=0;}
                    else
                        if(i==x&&j==y)h[i][j]=h[i][j-1];
                        else
                            h[i][j]=h[i+1][j]+h[i][j-1];
            if(h[1][m]<nrmin&&h[1][m]!=0)nrmin=h[1][m],xmin1=x,ymin1=y,nr=1;
            else
                if(h[1][m]==nrmin)nr++,xmin2=x,ymin2=y;}
    cout<<nrmin<<' '<<nr<<'\n';
    cout<<xmin1<<' '<<ymin1-1<<' '<<xmin1<<' '<<ymin1<<'\n';
    cout<<xmin2<<' '<<ymin2-1<<' '<<xmin2<<' '<<ymin2<<'\n';
    return 0;
}
```

The variable *nrmin* returns the minimum number of paths, and the pair of variables *(xmin1,ymin1)* and *(xmin2,ymin2)* carry the coordinates of the final nodes of the searched streets.

For *n=4* and *m=6*, we obtain that the minimum number of paths is *21*. There are *two* streets with this property, namely *the street that connects nodes* $(1,5)$ *and* $(1,6)$ and *the street that connects nodes* $(4,1)$ *and* $(4,2)$.

Second method: A computing method

The number of paths that *don't contain* a given street is minimal if and only if the number of paths *containing* that street is maximal. For each street *CD* of the map, we count how many paths contain that street. A path through *CD* is composed by a path from *A* to *C*, street *CD* and a path from *D* to *B*.

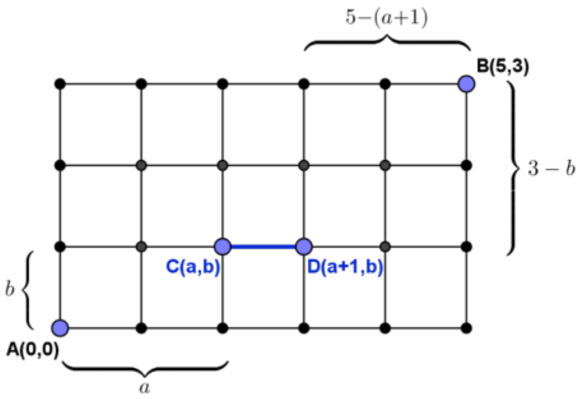


Figure 9

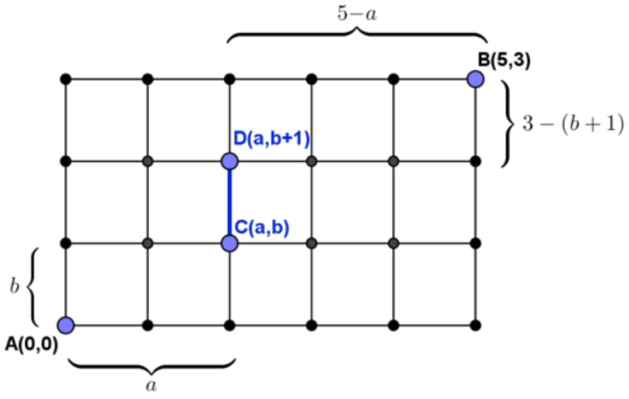


Figure 10

i) If the street is horizontal and joins point *C*, of coordinates (a,b), with point *D*, of coordinates $(a+1,b)$, where $0 \le a \le 4$, $0 \le b \le 3$ (see Figure 9), then the map of paths from *A* to *C* is a $a \times b$ rectangle and the map of paths from *D* to *B* is a rectangle with sides of length $5-(a+1)=4-a$ and $3-b$. Therefore, by *the product rule*, the number of paths that contain street *CD* is

$$D_1(a,b)=\binom{a+b}{a}\cdot\binom{(4-a)+(3-b)}{4-a}=\frac{(a+b)!}{a!\cdot b!}\cdot\frac{(7-a-b)!}{(4-a)!\cdot(3-b)!}.$$

ii) If the street is vertical and joins point *C*, of coordinates (a,b), with point *D*, of coordinates $(a,b+1)$, where $0 \le a \le 5$, $0 \le b \le 2$ (see Figure 10), than the map of paths from *A* to *C* is a $a \times b$ rectangle and the map of paths from *D* to *B* is a rectangle with sides of length $5-a$ and $3-(b+1)=2-b$. It follows that the number of paths that contain street *CD* is

$$D_2(a,b)=\binom{a+b}{a}\cdot\binom{(5-a)+(2-b)}{5-a}=\frac{(a+b)!}{a!\cdot b!}\cdot\frac{(7-a-b)!}{(5-a)!\cdot(2-b)!}.$$

We are now investigating the position and the orientation of the street *CD* for which the number of paths passing through this street is maximum. We have the following cases:

- If $b=0$, then

$$D_1(a,0)=1\cdot\frac{(7-a)!}{(4-a)!\cdot 3!}=\frac{(5-a)(6-a)(7-a)}{6};$$

$$D_2(a,0)=1\cdot\frac{(7-a)!}{(5-a)!\cdot 2!}=\frac{(6-a)(7-a)}{2}.$$

Because expressions $5-a, 6-a, 7-a$ get larger as a gets smaller, the maximum values in this case are $D_1(0,0)=35$ and $D_2(0,0)=21$; we keep in mind the highest value, $D_1(0,0)=35$.

- If $b=1$, then

$$D_1(a,1)=\frac{(a+1)!}{a!\cdot 1!}\cdot\frac{(6-a)!}{(4-a)!\cdot 2!}=\frac{(a+1)(5-a)(6-a)}{2};$$

$$D_2(a,1)=\frac{(a+1)!}{a!\cdot 1!}\cdot\frac{(6-a)!}{(5-a)!\cdot 1!}=(a+1)(6-a).$$

From the table of values

a	0	1	2	3	4	5
$D_1(a,1)$	15	20	18	12	5	-
$D_2(a,1)$	6	10	12	12	10	6

we reason that in this case D_1 and D_2 take on values smaller than 35.

- If $b=2$, then

$$D_1(a,2)=\frac{(a+2)!}{a!\cdot 2!}\cdot\frac{(5-a)!}{(4-a)!\cdot 1!}=\frac{(a+1)(a+2)(5-a)}{2}$$

and the table of values

a	0	1	2	3	4
$D_1(a,2)$	5	12	18	20	15

shows, again, smaller values than 35. Moreover,

$$D_2(a,2)=\frac{(a+2)!}{a!\cdot 2!}\cdot 1=\frac{(a+1)(a+2)}{2},$$

and the expressions $a+1, a+2$ get larger as a gets larger, hence the highest value that we can obtain is $D_2(5,2)=21<35$.

- If $b=3$, then

$$D_1(a,3)=\frac{(a+3)!}{a!\cdot 3!}\cdot 1=\frac{(a+1)(a+2)(a+3)}{6},$$

and the expressions $a+1, a+2, a+3$ increase as a increases, so the maximum value in this case is $D_1(4,3)=35$.

In conclusion, the largest number of paths, 35, pass either through the street that joins the points of coordinates $(0,0)$ and $(1,0)$ or through the street that joins the points of coordinates $(4,3)$ and $(5,3)$ (the problem has *two* solutions – see Figure 11).

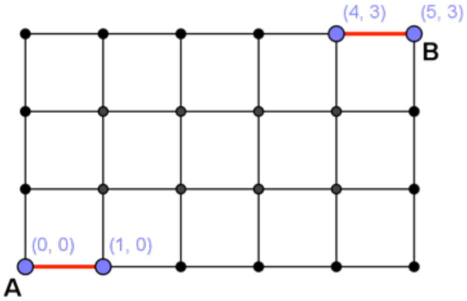


Figure 11

We observe that the street that joins the points of coordinates $(0,0)$ and $(1,0)$ is the street that joins the nodes $(4,1)$ and $(4,2)$ from the C++ program (see Figure 4). Also, the street that joins the points of coordinates $(4,3)$ and $(5,3)$ is the street that joins the nodes $(1,5)$ and $(1,6)$ from the program.

If one of these streets closes, the number of ways of driving from A to B is minimal. More precisely, since the total number of paths that join A and B is 56, there are $56-35=21$ possible paths left, as the C++ program also states.

On two of the *Big Hill* town streets (as marked in Figure 12) access is forbidden for cars and pedestrians alike.

(b) **In how many ways can one drive from point A to point B of *Big Hill*, taking into account the restrictions?**

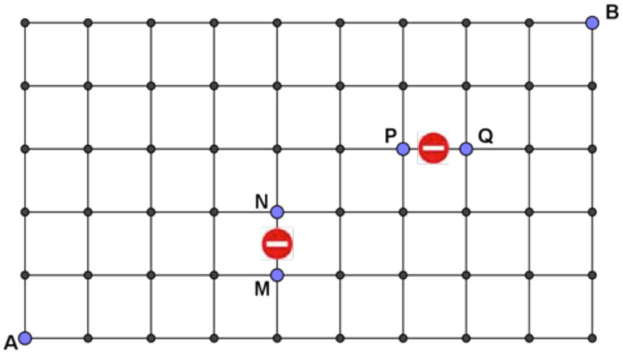


Figure 12. The center of the town *Big Hill*

First, since the map in Figure 12 is a rectangle with the longer side of 9 streets and the shorter side of 5 streets, proceeding as at point (a), we infer that if none of the streets had been closed to traffic, the number of possible paths from A to B would have been

$$D_{AB}=\binom{9+5}{5}=\frac{14!}{9!\cdot 5!}=\frac{10\cdot 11\cdot 12\cdot 13\cdot 14}{1\cdot 2\cdot 3\cdot 4\cdot 5}=2002.$$

In order to calculate how many paths avoid streets MN and PQ, we will subtract from D_{AB} the number D_{closed} of paths that contain at least one of the streets MN and PQ. According to *the inclusion-exclusion principle*, the value D_{closed} is the sum of the number D_{MN} of paths from A to B that contain street MN and the number D_{PQ} of paths from A to B that contain street PQ, from which we subtract the number $D_{MN,PQ}$ of paths from A to B that contain both streets, because these paths have been summed twice.

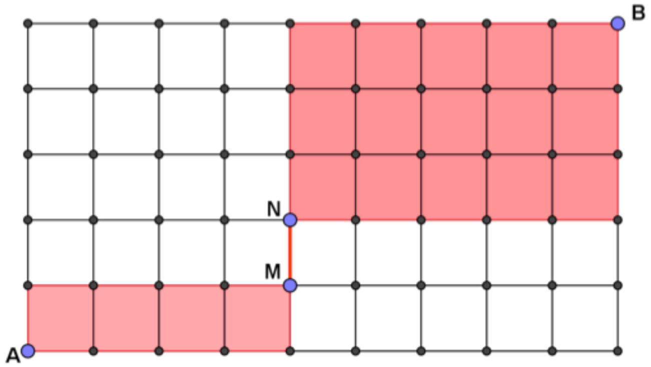


Figure 13

A path from A to B including street MN is the union of a path from A to M, street MN and a path from N to B (see Figure 13). Because the map of the streets from A to M is a 4×1 rectangle, there are $\binom{4+1}{1}=\binom{5}{1}=5$ possible paths A-M. Since the map of the streets from N to B is a 5×3 rectangle, there are $\binom{5+3}{5}=\frac{8!}{5!\cdot3!}=\frac{6\cdot7\cdot8}{6}=56$ admissible paths N-B. Therefore, $D_{MN}=5\cdot1\cdot56=280.$

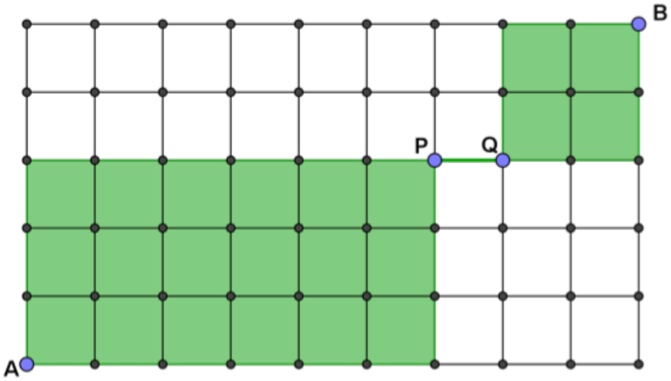


Figure 14

A path from A to B via street PQ is the union of a path from A to P, street PQ and a path from Q to B. As the map of streets from A to P is a 6×3 rectangle, and the map of streets from Q to B is a square with side length 2 (see Figure 14), we obtain analogously that

$$D_{PQ}=\binom{6+3}{6}\cdot1\cdot\binom{2+2}{2}=\binom{9}{6}\cdot\binom{4}{2}=\frac{9!}{6!\cdot3!}\cdot\frac{4!}{2!\cdot2!}=\frac{7\cdot8\cdot9}{6}\cdot\frac{3\cdot4}{2}=504.$$

Finally, a path from A to B containing streets MN and PQ is the union of a path from A to M, street MN, a path from N to P, street PQ and a path from Q to B. Because the map of the streets from A to M is a 4×1 rectangle, the map of the streets from N to P is a 2×1 rectangle, while the map of the streets from Q to B is a square with side length 2 (see Figure 15), we get

$$D_{MN,PQ}=\binom{4+1}{1}\cdot1\cdot\binom{2+1}{1}\cdot1\cdot\binom{2+2}{2}=\binom{5}{1}\cdot\binom{3}{1}\cdot\binom{4}{2}=5\cdot3\cdot\frac{4!}{2!\cdot2!}=15\cdot\frac{3\cdot4}{2}=90.$$

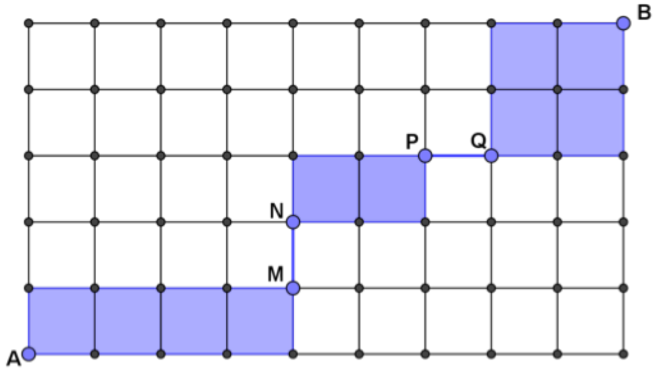


Figure 15

Consequently, $D_{closed}=D_{MN}+D_{PQ}-D_{MN,PQ}=280+504-90=694$, hence the number of paths from A to B which avoid the restricted streets are $D_{AB}-D_{closed}=2002-694=1308$.

(c) **Two walkers, *Alin* and *Bianca*, depart simultaneously from points A and B, respectively. *Alin* may move only one square right or upwards, while *Bianca* may move only one square left or downwards (when looking at the grid from the same perspective). For each of them, the choice of direction is made with equal probabilities in every intersection. What are the odds that *Alin* and *Bianca* meet?**

Because the map of the paths from A to B is a 9×5 rectangle, in order to get from A to B, no matter what path he may choose, *Alin* must walk 9 streets to the right and 5 streets upwards, that is $9+5=14$ streets. Likewise, in order to get from B to A, *Bianca* must also walk 14 streets (9 to the left and 5 downwards). **Supposing they walk at the same speed**, they must meet halfway, that is, after each of them walked $14:2=7$ streets.

Alin can walk the 7 streets in 2^7 equally possible ways, because from every node of the grid he has two equally possible ways of choosing the direction. Analogously, *Bianca* also has 2^7 equally possible ways of walking 7 streets. Hence, by *the product rule*, the pair *Alin-Bianca* may move along $2^7\cdot2^7=2^{14}$ equally possible trajectories.

On the other hand, as they meet, we can merge their trajectories and we get a path that connects points A and B, travelling to the right or upwards. In other words, the set of the favorable cases is the set of all paths connecting A and B, on the conditions of the problem. The number of favorable cases is $\binom{14}{5} = 2002$, thus the probability that *Alin* and *Bianca* meet is equal to $\frac{2002}{2^{14}} = \frac{1001}{2^{13}} = \frac{1001}{8192}$; $0{,}122$. Otherwise stated, the odds that *Alin* and *Bianca* meet are approximately 1:7.

(d) **Apart from the two above mentioned restrictions, other streets in the town will be under repair and will be inaccessible for a while. Find out the maximum number of streets that can be under repair simultaneously, so that all the touristic objectives are connected to point *A* by at least one path. We maintain the original rules of movement. The path starting at *A* may end in any point (touristic objective).**

We count how many streets we may close if we want to reach from *A* to any other node, not necessarily getting to *B*.

Node N_1, the bottom right corner of the rectangle, can be reached only by following the bottom edge of the map, hence the streets on this edge cannot be under repair. At its turn, node N_2, the upper-left corner of the rectangle, can be reached only by following the left vertical edge of the map, hence the streets on this edge cannot be under repair either (see Figure 16). In this way, we obtain that *every* node on these two edges can be reached by a path starting at *A*.

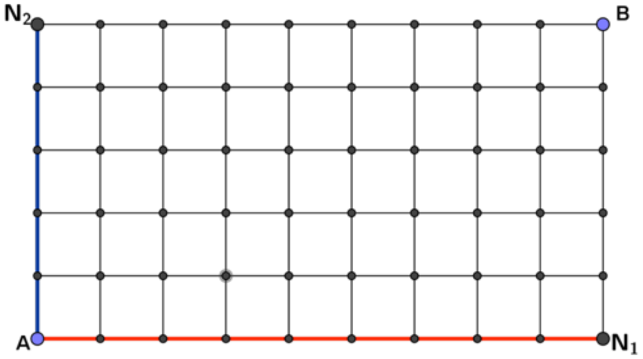


Figure 16

The rest of the nodes on the map can be reached from two directions (namely, from the left and from below), hence if we keep one of these streets, we may have at least a path from *A* to each of these nodes. Since the grid has 10 nodes on the horizontal and 6 nodes on the vertical, by removing the $10+6-1=15$ nodes on the left and bottom sides, there are $6 \cdot 10 - 15 = 45$ nodes left. In conclusion, in this case we may close at most 45 *streets*.

An example of streets closure is given in Figure 17.

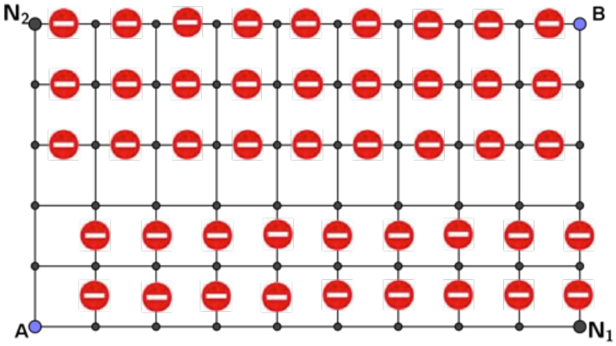


Figure 17

5. CONCLUSIONS

We have answered all the questions completely. We have also added suplimentary tasks to the problem, such as the C++ programming and different approaches to the solution. By studying this problem, we have learned more techniques in combinatorics, probability theory and their applications. The problem could be a starting point in solving more delicate problems, such as the traffic light coordination problem in busy city centres.

6. REFERENCES

1. J. Batterson, *Competition Math For Middle School*, AoPS Inc., 2011.
2. D. Fomin, S. Genkin, I. Itenberg, *Mathematical Circles (Russian Experience)*, Universities Press, 1996.