Counting Configurations

[Year 2022– 2023]

FIRST NAMES AND SURNAMES OF STUDENTS, GRADES: ALEXANDRA BURBULEA, SORIN-GABRIEL GABĂRĂ, ALEXANDRU IONIȚĂ, MIHAI MAXIM, RĂZVAN TĂNASĂ, ADRIAN SOLCANU, VICTOR SOLCANU, students in 8th grade

SCHOOL: Colegiul Naţional *C. Negruzzi*, Iaşi

TEACHER: ADRIAN ZANOSCHI, Colegiul Național *C. Negruzzi* and GEORGE-IOAN STOICA, student at l'Ecole Polytechnique Paris

RESEARCHER: IULIAN STOLERIU, Faculty of Mathematics, *Al. I. Cuza* University, Iași

PRESENTATION OF THE RESEARCH TOPIC

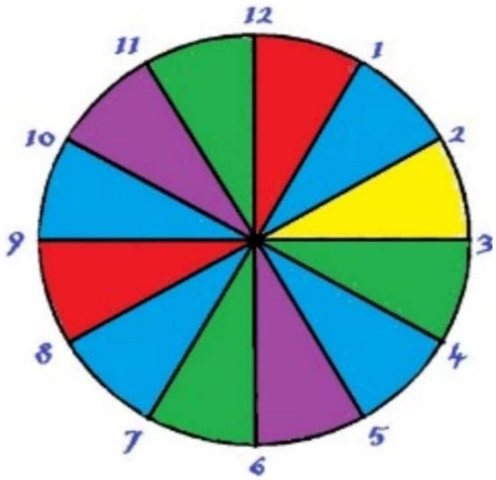
Problems that require determining the number of configurations of a possible map are essential for combinatorial problems and algorithms for computer programming. We want to present how many combinations of painting the sectors of the circle are possible, if the adjacent colours cannot be the same. We will solve this in the particular case of 12 sectors and 5 colours and then we will find a generalisation. Moreover, we want to find out the number of configurations (which respect the same conditions) without rotations .

BRIEF PRESENTATION OF THE CONJECTURES AND RESULTS OBTAINED

The 12 hour sectors of an analogue clock are to be painted in such a way that each sector is painted by one colour and there are no two adjacent sectors having the same colour. If there are 5 colours available, in how many ways can the painting be done? Generalize for n sectors and k available colours.

Imagine now a big circular pizza that is cut into 12 equal slices and 5 available toppings. The pizza must be topped in such a way that each slice has a different topping and there are no two adjacent slices having the same topping. In how many ways can this be done? Generalize to n slices and k toppings.

This article is written by students. It may include omissions and imperfections, as far as possible, reported by our reviewers in the editorial notes.



3. **SOLUTION**

The first task is to find a general formula for the number ways to colour n sectors by $\underline{k}$ colours. We shall get result this by using two methods.

Method 1. Our first idea was to try and calculate the number of possibilities for smaller and trivial cases.

Also, these are some rules that we shall obtain (for n - number of sectors and k - number of colours):

1. The number of sectors should be equal or greater than the number of colours used in order to have at least one configuration.

2. $k \geq 2$, so that the number of configurations is larger than 1;

3. If $k = 2$, then the number of configurations is always even;

Proof:

Let's suppose that $k = 2$ and n -odd:

n - odd $\Rightarrow n = 2x + 1$, $x \in \mathbb{N}^*$

$k = 2$: colour a and colour b

We would have:

Sector 1: colour a

Sector 2 : colour b

Sector 3 : colour a

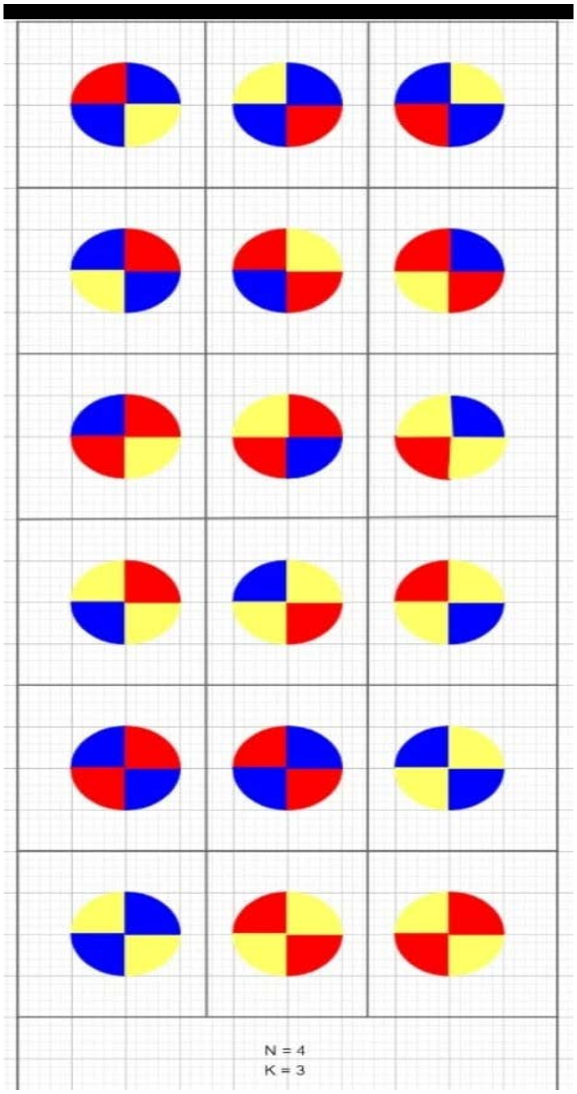
…

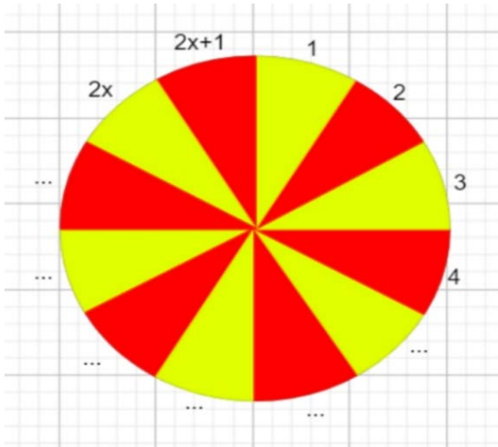
Sector $2x - 1$: colour a

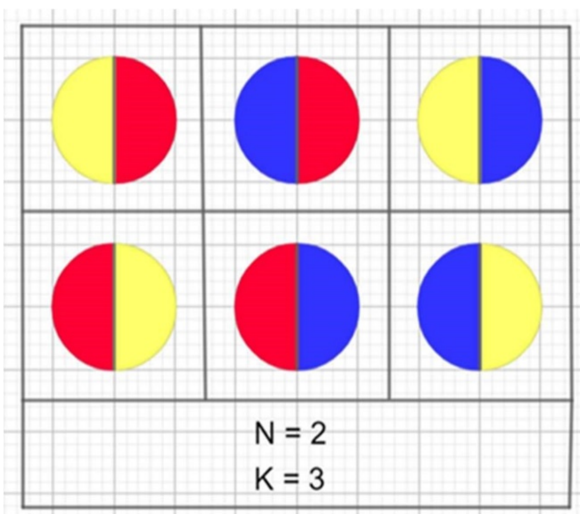
Sector $2x$: colour b

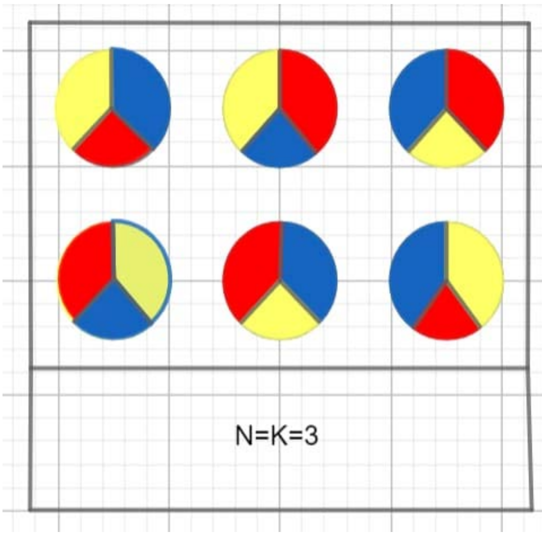
Sector $2x+1$: colour a

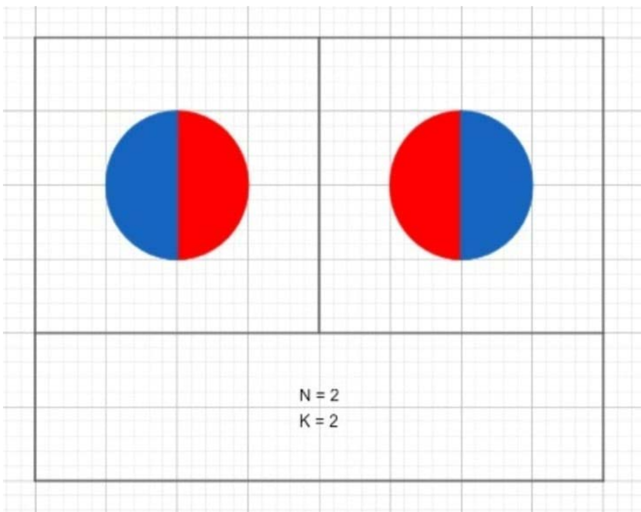
But, sector $2x+1$ is adjacent to sector 1 ⇒ they can't have the same colour⇒ if $k=2$, $n \neq 2x+1$⇒ n -even.




2x+1
1
2x
2
3
4
...


N = 2
K = 3





As we have observed in the earlier part of our presentation, we must determine in how many ways can we colour the circle so that no two adjacent sectors have the same colour. So, we will let $a(n,k)$ be the number of configurations for n sectors and k colours.

In the case where $n = 12$ and $k = 5$, we may assume that the first sector can be coloured in 5 ways and the other ones in 4 ways, because they cannot get the colour of the previous sector. Hence, the result should be $5 \cdot 4^{11}$. However, this is not the right answer, as we still take in consideration some configurations where the last and first have the same colour.

That's why we thought that we can find a way to subtract those cases. The easiest way to do it is to consider the last and first sector as only one unit. This means we must count the valid configurations for a clock with 11 sectors and 5 colours.

However, this can be written as $a(11,5)$. Actually, you can immediately see the process is repetitive and it goes until you reach $a(1,5)$, equal to 0 (We know that n and k have to be higher or equal to2).

Hence, we have $a(12,5) = 5 \cdot 4^{11} - a(11,5)$;

$$a(11,5) = 5 \cdot 4^{10} - a(10,5);$$
$$a(10,5) = 5 \cdot 4^{9} - a(9,5);$$
$$\ldots$$
$$a(1,5) = 0.$$

If we start calculating, it means that:

$$a(2,5) = 5 \cdot 4^1 - 0;$$
$$a(3,5) = 5 \cdot 4^2 - (5 \cdot 4^1 - 5) = 5 \cdot 4^2 - 5 \cdot 4^1 + 0;$$
$$a(4,5) = 5 \cdot 4^3 - (5 \cdot 4^2 - 5 \cdot 4^1 + 5) = 5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1 - 0;$$
$$a(5,5) = 5 \cdot 4^4 - (5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1 - 5) = 5 \cdot 4^4 - 5 \cdot 4^3 + 5 \cdot 4^2 - 5 \cdot 4^1 + 0;$$

We can easily observe that the signs alternate, depending on the parity of n (even or odd). The number of configurations for the case of $n = 12$ and $k = 5$ will look like this:

$$a(12,5) = 5 \cdot 4^{11} - 5 \cdot 4^{10} + 5 \cdot 4^9 - 5 \cdot 4^8 + \ldots + 5 \cdot 4^1 - 0$$

If we subtract $k = 5 = 5 \cdot 4^0$, we have $a(12,5) - k =$

$$= 5 \cdot 4^{10}(4-1) + 5 \cdot 4^8(4-1) + 5 \cdot 4^6(4-1) + 5 \cdot 4^4(4-1) + 5 \cdot 4^2(4-1) + 5 \cdot 4^0(4-1) =$$
$$= 15(4^{10} + 4^8 + 4^6 + 4^4 + 4^2 + 1)$$

However, we can see that, if try to write a simplified form for a case with an odd number of sectors, we won't need to subtract k anymore. As an example, we can take a(5,5). So, we have:

$$a(5,5) = 5 \cdot 4^4 - (5 \cdot 4^3 - 5 \cdot 4^2 + 5 \cdot 4^1 - 5) = 5 \cdot 4^4 - 5 \cdot 4^3 + 5 \cdot 4^2 - 5 \cdot 4^1 + 0 =$$

$$a(5,5) = 5 \cdot 4^3(4-1) + 5 \cdot 4^1(4-1) = 15 \cdot (4^3 + 4^1) = 15 \cdot 4 \cdot (4^2 + 1)$$

In order to narrow down the number of configurations even further, we can take another two subcases: one for an even value of n, and another one for an odd value of *n*.

Case I: n-even $(n = 2x)$

We get that:

$$(k-2) \cdot k \cdot \left[(k-1)^{2x-2} + (k-1)^{2x-4} + \ldots + (k-1)^{2x+1-(2x+1-3)} + (k-1)^{2x-(2x-2)} + (k-1)^0\right] + k =$$

$$= (k-2)\cdot k\cdot\left\{[(k-1)^2]^{x-1} + [(k-1)^2]^{x-2} + \ldots + [(k-1)^2]^{x-(x-1)} + [(k-1)^2]^0\right\} + k =$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^x - 1}{(k-1)^2 - 1} =$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^x - 1}{(k-1-1)(k-1+1)} + k =$$
$$= (k-2)\cdot k\cdot\frac{[(k-1)^2]^{\frac{n}{2}} - 1}{(k-2)\cdot k} + k =$$
$$= [(k-1)^2]^{\frac{n}{2}} - 1 + k =$$
$$= (k-1)^n - 1 + k$$

Case II: n-odd ($n = 2x + 1$)

We get that:

$$(k-2)\cdot k\cdot\left[(k-1)^{2x+1-2} + (k-1)^{2x+1-4} + \ldots + (k-1)^{2x+1-(2x+1-3)} + (k-1)^{2x+1-(2x+1-1)}\right] =$$
$$= (k-2)\cdot k\cdot(k-1)\times[(k-1)^{2x-2} + (k-1)^{2x-4} + \ldots + (k-1)^2 + (k-1)^0] =$$

$$= (k-2) \cdot k \cdot (k-1)\{[(k-1)^2]^{x-1} + [(k-1)^2]^{x-2} + \ldots + [(k-1)^2]^1 + [(k-1)^2]^0\} =$$
$$= (k-2) \cdot k \cdot (k-1) \cdot \frac{[(k-1)^2]^x - 1}{(k-1)^2 - 1} =$$
$$= (k-2) \cdot k \cdot (k-1) \cdot \frac{[(k-1)^2]^x - 1}{(k-1-1) \cdot (k-1+1)} =$$
$$= (k-2) \cdot k \cdot (k-1) \cdot \frac{[(k-1)^2]^x - 1}{(k-2) \cdot k} =$$
$$= (k-1) \cdot \{[(k-1)^2]^x - 1\} =$$
$$= (k-1) \cdot \left\{[(k-1)^2]^{\frac{n-1}{2}} - 1\right\} =$$
$$= (k-1) \cdot [(k-1)^{n-1} - 1] =$$
$$= (k-1) \cdot (k-1)^{n-1} - (k-1) =$$
$$= (k-1)^{1+n-1} - k + 1 =$$
$$= (k-1)^n - k + 1$$

In conclusion, the formulae that we have obtained are:

I. $(k-1)^n - k + 1$, if *n* is odd.

II. $(k-1)^n - 1 + k$, if *n* is even.

The general formula for both cases (n-even or n-odd) can be written as:

$$\boldsymbol{a(n,k) = (k-1)^n + (-1)^n \cdot (k-1)}$$

In particular, for $n = 12$and $k = 5$, we get $a(12,5) = 4^{12} + 4 = 16777220$ configurations and for n=12 and k=8, we get a(12,8)=13841287208 configurations

```
from operator import eq
from itertools import product


def anyeq(a):
    a.append(a[0])
    return not any(map(eq, a, a[1:]))


colors = [1, 2, 3, 4, 5]

combi = product(colors, repeat=12)
print(len([1 for x in combi if anyeq(list(x))]))
```

This was the first code we have written. Although it is very slow, because it counts the configurations one at a time, it helped us find the values for smaller numbers, before finding the formulas.

Now, we are going to move on to the C++ solution for the first part of the problem.This code was creating by converting the formulas obtained above into the C++ programming language.

The C++ code helps us:

1) find values of higher pairs of numbers;
2) reduce the time in which the solutions can be found;
3) reduce the chance of an error occurring;

This is the C++ code for the first part of the problem. We chose a set of variables to help us find the solution: k – the number of colours; n– the number of sectors in which the analogue clock will be divided;

The code only works if the number of colours and the numbers of sectors are larger than 1;

```
#include <iostream>

using namespace std;

int main()
{
    int k,n,i,x;
    cout<<"Introduce the number of sectors: ";
    cin>>n;
    cout<<"Introduce the number of colours: ";
    cin>>k;
    x=1;
    if(n%2==0)
    {
        for(i=0;i<n;i++)
        {
            x=x*(k-1);
        }
        cout<<"The number of solutions is: "<<x-1+k;
    }
    else
    {
        for(i=0;i<n;i++)
        {
            x=x*(k-1);
        }
        cout<<"The number of solutions is: "<<x+1-k;
    }
    return 0;
}
```

We chose to divide the problem into two separate parts:

- **I.** When n is divisible by 2 (even)
- **II.** When the remainder of n divided by 2 is 1 (odd)

I. When n is divisible by 2 (even; $2|n$).

```
if(n%2==0)
{
    for(i=0;i<n;i++)
    {
        x=x*(k-1);
    }
    cout<<"The number of solutions is: "<<x-1+k;
}
```

Within these lines we want to reach the formula:

$$(k-1)^n + k - 1$$

To do that, we need to choose a variable x, which is initially equal to 1, and another variable i that starts at 0 and will increase by 1, until it reaches n. Every time i increases, x is multiplied by $(k-1)$**,** which is the correspondent of the line $x = x \cdot (k-1)$;

Finally, outputs computed by the code are shown in a tab looking like this:

```
Introduce the number of sectors: 12
Introduce the number of colours: 5
The number of solutions are: 16777220
Process returned 0 (0x0)   execution time : 4.638 s
Press any key to continue.
```

SLIDE 17

II. When is not divisible by 2 (*n* is odd)

```
else
{
    for(i=0;i<n;i++)
    {
        x=x*(k-1);
    }
    cout<<"The number of solutions are: "<<x+1-k;
```

Within these lines we want to reach the formula:

$$(k-1)^n - k + 1$$

To do that, we need to choose a variable x, which is initially equal to 1, and another variable i that starts at 0 and will increase by 1, until it reaches n . Every time i increases, x is multiplied by$(k-1)$, which is the correspondent of the line $x = x \cdot (k-1)$;

Finally, outputs computed by the code are shown in a tab looking like this:

```
Introduce the number of sectors: 12
Introduce the number of colours: 5
The number of solutions are: 16777220
Process returned 0 (0x0)   execution time : 4.638 s
Press any key to continue.
```

$I. n = k = 2 \Rightarrow 2$ possibilities
$II. n = 2, k = 3 \Rightarrow 6$ possibilities
$III. n = k = 3 \Rightarrow 6$ possibilities
$IV. n = 4, k = 3 \Rightarrow 18$ possibilities
$V. n = 5, k = 3 \Rightarrow 30$ possibilities
$VI. n = 5, k = 4 \Rightarrow 240$ possibilities
$VII. n = 5, k = 5 \Rightarrow 1020$ possibilities
$VIII. n = 6, k = 3 \Rightarrow 66$ possibilities
$IX. n = 6, k = 4 \Rightarrow 732$ possibilities
$X. n = 6, k = 5 \Rightarrow 4100$ possibilities
$XI. n = k = 6 \Rightarrow 15630$ possibilities
$XII. n = 8, k = 3 \Rightarrow 258$ possibilities
$XIII. n = 8, k = 4 \Rightarrow 6564$ possibilities
$XIV. n = 9, k = 3 \Rightarrow 510$ possibilities
$XV. n = 9, k = 4 \Rightarrow 19680$ possibilities
$XVI. n = 9, k = 5 \Rightarrow 262140$ possibilities
$XVII. n = 9, k = 6 \Rightarrow 1953120$ possibilities

Method 2: Another way of proving the general formula written above. Let us denote by a_n the number of ways to colour the n sectors by k colours, satisfying the requirements in the problem. Firstly, let us imagine that the sectors numbered by 1 and *n* are not neighbours. Then, the total number of ways of colouring the *n* sectors by *k* colours, satisfying the requirements in the problem is $k \times (k-1)^{n-1}$. However, we can count all these possibilities in two ways:

SLIDE 20

1. The cases in which sectors 1 and *n* have different colours, which are equal to a_n.

2. The cases in which sectors 1 and *n* have the same colours, which are equal to a_{n-1}.

Therefore, we arrive at the following recurrence relation:

$$a_n + a_{n-1} = k \times (k-1)^{n-1}, \text{ for } n = 3, 4, \ldots, \text{ and } k = 3, 4, \ldots$$

The general term of the sequence a_n that solves the above recurrence is

SLIDE 21

B. In the second part we will have to count the number of different configurations that can be achieved with n sectors and k colours. The difference between the two parts is that here the order in which the different rotations that exist in every circle count as one case. At first, we discovered a formula, based on 2 observations and some further deduction:

1- Every time that n is prime, no configuration of colours repeats itself after a number of rotations.
2- From observation 1 we can deduce that only when we use a prime number x_1 of colours (divizible by n), the configuration repeats itself after a number of rotations.
SLIDE 22

Based on observation 2, we can subtract these configurations and add the correct ones later. After we get rid of these arrangements, we can divide the rest by n, the number of possible rotations. At last, we add the correct cases that we subtracted earlier and we can easily do this by dividing these configurations by $\frac{n}{x_1}$, the amount of times that they repeat themselves.

The formula can be written as:

$$b(n,k)n = \frac{a(n,k) - \frac{k!}{(k-x_1)!} - \frac{k!}{(k-x_2)!} - \ldots - \frac{k!}{(k-x_{\tau_n-2})!}}{n} + \frac{k!}{(k-x_1)! \cdot \frac{n}{x_1}} + \frac{k!}{(k-x_2)! \cdot \frac{n}{x_2}} + \ldots + \frac{k!}{(k-x_{\tau_n-2})! \cdot \frac{n}{x_{\tau_n} - 2}}$$

Unfortunately, this only works for the cases where

$$k \geq \frac{n}{\text{his smallest prime divizor}}$$

SLIDE 23

This lower bound should help us to get the second formula:

For the particular case n=6, k=4, we computed the number of configurations as it follows:

We assume that the four colours are: A, B, C and D.

- **If two colours are used,** we will have: 6 possibilities (e.g., 1 possible arrangement is ABABAB), which are the 6 different groups of two colours (AB, AC, AD, BC, BD, CD)).
- **If three colours are used,** we will have:
    I. Using 1A, 2B, 3C , the only possible arrangement is: CBCBCA.
    4 (ways to choose the 3 colours) · 6 (permutations)=24 possibilities

II. Using 2A, 2B, 2C, the possible arrangements are: ABCABC, ABACBC, ABACBC, ABCACB and ACBACB.
$4\text{ (ways to choose the 3 colours)} \cdot 5\text{ (arrangements)}=20\text{ possibilities}$

Total (for three colours used): $24+20=44$ possibilities.
SLIDE 24

- **If four colours are used,** we will have:

I. Using 1A, 1B, 1C, 3D, the possible arrangements are: ADBDCD and ADCDBD.
$$2\text{ (arrangements)} \cdot 4\text{ (permutations)}=8\text{ possibilities}$$

II. Using 1A, 1B, 2C, 2D, we can choose in six ways which colour to be used twice.
$$\frac{5!}{1!\,2!\,2!}=30\text{ ways to arrange ABCCDD}$$
$$\frac{4!}{1!2!2!}\cdot 2=24\text{ cases when two sectors colored in C or D are adjacent } \frac{3!}{1!1!1!}=6\text{ cases when two sectors colored in C are adjacent and two sectors colored in D are adjacent}$$

SLIDE 25

The total number of configurations when four colours are used is

$$8+6\cdot(30-24+6)=$$
$$=8+6\cdot 12=$$
$$=80\text{ possibilities}$$

Total = $6+44+80=130$ possibilities

For n=12 and k=8 there are b(12,8)=1153443844 configurations

SLIDE 26

4. CONCLUSION

To solve this problem, we only needed to use calculus. In order to find the formulas that we used to generalize the problem for *n* sectors and *k* colours, we analysed a few particular cases to grasp an idea about how to approach the problem. Also, we have written a C++ code to determine the solutions for higher cases.

To solve the second part of the problem, we tried to find a formula based on the number of divisors of n. Also, we calculated a particular case to verify the formula.

To reach the final formula, we used the results found during the solution.

SLIDE 27