Matches

Imagination sometimes is more important than knowledge

Jeyhun Tahirov jeyhunt@42.usa.com

Summary: This project helps developing skills required to create your own ***`algorithms`***

Contents

I **Foreword** **2**

II **Introduction** **3**

III **General instructions** **4**

IV **Matches** **5**

V **Bonus part** **6**

V.0.1 3D instead of 2D 6

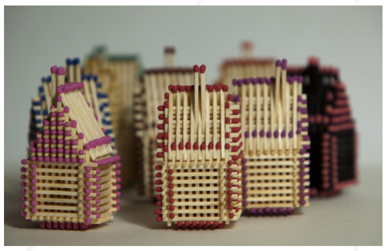
V.0.2 Time 6

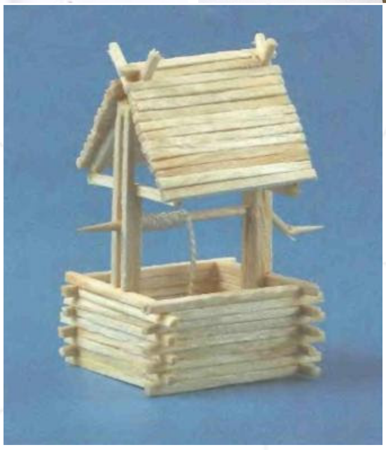
VI **Turn-in and peer-evaluation** **7**

Chapter I

Foreword

Since this puzzle is about arranging matches in a specific manners here is what else you can do when you don't know how to solve this puzzle:





Chapter II

Introduction

This puzzle is all about algorithms. Don't be fooled, problems are actually easy to solve, however they will require some creativity from you. Try to use pen and paper to figure out the algorithm before you start coding.

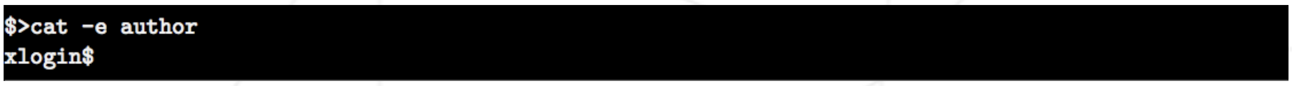
`Try to solve the puzzle by hand with pen and paper before code!`

Creating something new can be really hard if you are not using good old math. There are a lot of different ways how to solve this project. You can use math, or try to do dynamic modelling, but be aware in order to get bonuses you program should not take longer than 1 second.

Chapter III

General instructions

- This project will only be corrected by other human beings. You are therefore free to organize and name your files as you wish, although you need to respect some requirements below.
- The executable file must be named matches
- You'll have to submit at the root of your folder, a file called author containing your username followed by a '\n'

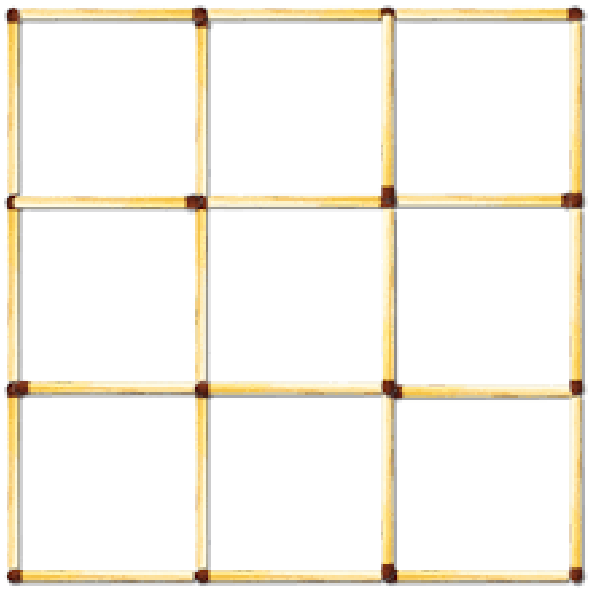


```
$>cat -e author
xlogin$
```

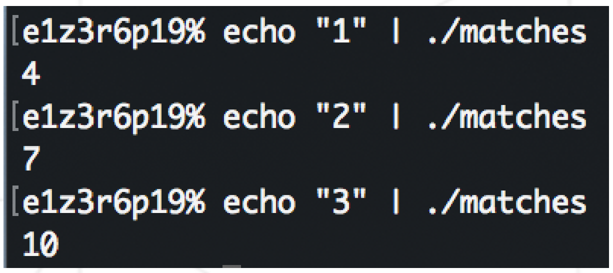
Chapter IV

Matches

What is the minimum number of matches necessary to put on the plane to create n squares with a side of one match? Matches can not be broken or placed on each other. The vertices of the squares should be the points where the ends of the matches meet, and the sides are matches themselves.



Write a program that read a number `n` from `STDIN` and prints to the `STDOUT` the `minimum amount of matches` required to construct `n squares`
This is how you should be able to test your program:



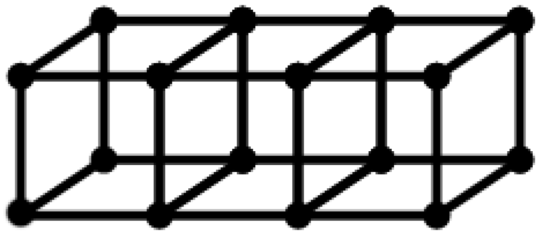
```
[e1z3r6p19% echo "1" | ./matches
4
[e1z3r6p19% echo "2" | ./matches
7
[e1z3r6p19% echo "3" | ./matches
10
```

Chapter V

Bonus part

V.0.1 3D instead of 2D

Write a separate program called `bonus` that will work in `3D`. It means the number `n` you are going to get are actually represent number of cubes to construct from matches not squares. For example if $n = 3$ then your program output should be `28`. `n` can not be higher than $2 * 10^9$



V.0.2 Time

You programs should be able to solve within 1 sec.
You can get this bonus twice. One for the mandatory part and other for the bonus part.
Good luck!

Chapter VI

Turn-in and peer-evaluation

Turn your work in using your `GiT` repository, as usual. Only work present on your repository will be graded in defense.