

## **Little info about our Fast Lotto Full Wheeling System**

First, thank you for your interest in this kind of software. I hope that this will be helpful to you to make lottery software for you or for the market and this, if you have lucky, can change your life.

Like you can see on screen when you start this „little“ application this is not END USER version. From that reason, we coded minimal GUI interface, with little or without at all checking on many user actions. This way application can be smaller (we talk about source code), so you can easily follow important things about combination generation, filtering with some conditions, how to cut down number of combinations... and other things about how to handle user interaction are left out because, you already know that.

I will try here to be short as possible and not to waste your time. Just to make a point about what you will find here.

This application is written in POWER BASIC. I know that some other programmers who use C/C++, NET, Delphi really don't like even to hear word BASIC. But believe me, it is not rare situation when they all use this compiler to make libraries (DLL) for their software, but they don't want you to know that. For many years this programmer tool is secret weapon for programmers, they use it, but if you ask them, they will never admit that. But, this is not so important. Important is that BASIC is easy to follow programming language, this tools generate very fast execution code, you can make DLL from them and just one more nice thing is that you can write part or all application using inline assembler, that you can find in it. We can debate, is this tool good or bad, but irrelevant. I don't want to push to this tool, I only want to show in this tools how some things can be done, so you can easily follow all, understand all and latter you can try to reprogram this in your programming language.

First thing is generating full wheel of numbers (in this application we talk about 7 numbers in combination from 8-39 numbers).

This can be easily changed to 6/49 if in your area 7/39 are not around.

You will see how fast your PC can generate combinations. On 2GHz PC time to generate 15380937 combination (full wheel 7/39) is about 0.06 seconds. On faster machines this time can be 0.03-0.04 seconds.

Maybe you don't believe that, but this is a fact. You will see short code inside source code. So, sometimes you don't need to allocate memory for all these combinations if you can generate them fast. Remember, to store all of this in array you will need  $15380937 * 7 * 4$  bytes. Why 4, because we are talking about 32bit PC so, it is much faster to access data in this form then to access one byte. This is very important to us. If you change size from LONG to BYTE you will save memory but you will lost speed.

If you don't believe that, try it for yourself and see.

Another story is when we need to store combinations in memory, RAM is much slower then processor registers, so we will reach 0.18 seconds to store 15380937 combinations in memory. This application do that to.

When we generate full wheel we can try to make abbreviated system, and you can see that. We will try to find small number of combinations that will guaranty us 5 or 4 from 7. There is many abbreviated systems, and we will try to find one as small as we can.

When we deal with all 39 numbers, we can set some conditions about sum of numbers in every combinations, number of odd/even numbers, one our lucky number that we want in every combinations and we can set range for some numbers. You will see how fast it can be done. How we cat number of combinations down. When you accept all what you find, here only your imagination is limit for what you can add as condition.

For example maybe you want that in combinations you never see 3 or 4 one by one numbers. Maybe you want to choose only combination without numbers from last draw, maybe you want to test some systems from others, maybe you want more then one abbreviated systems together and test what will happened,. Maybe you want to test what will happen if you have good guess on only 6 or 5 numbers. All of it you can add by yourself and you can make software with your lottery logic in it. Now you have speed under your hands.

Be careful if you choose option to write to disk all generated combination for some systems. Speed of writing is about 10000 combination per second. Size can be up to 500MB. On disk you will find first number of combination, then seven numbers, all data are comma separated for every combination, so you can use this TXT file if you need this kind of information. We didn't add text file viewer, application will start wordpad from windows accessory. If you don't have it, or it is not in the right directory you will not see text file. In that situation use your own text file viewer. We just want that this application be as simple as possible. GUI is not of interest here,

and interaction with user is irrelevant too. We add this only a little as we need to show you what we are talking about, you have to make your own nice looking screens with good control, but once again, we believe that you already know how to do that.

You will get one BAS file which is about 6000 lines. I have tried to do my best and comment all of it so you can easily follow even assembler code. If you know nothing about assembler, maybe it's time to learn a little, believe me it is not as hard as you think and there is no reason not to try to add some SPEED to some of your applications.

**Warning** when you choose to find abbreviated system with gar. 4 or 5 numbers of 7 good ones, be very careful. This part is not fully written in assembler in this demo, so duration of this task can be very long. Why? Like we said, only one small part is written in assembler, second, we will test as many abbreviated systems as we have number of combinations because we want to choose the best one (read smallest one). For example if you have 10000 combinations we will make 10000 abbreviated systems, for each of them we must check 1000 combinations and build abbreviated one and at the end we will choose one with smallest number of combinations.

If you have questions, feel free to ask. I will try to answer.

Use my email [alek@as-computer-software.com](mailto:alek@as-computer-software.com)

Wish you all the best

Aleksandar Stefanovic  
AS Computer Software  
January 2011.