Project Java

Jonathan Lee

Sierra College

Fall 2021 Tutor Center

Oct 1st 2021

Java CSCI12 Peer Tutoring Project

Kyptos

==========

Lets try to decode Kyptos with a cipher wheel program. Keyword being Try.

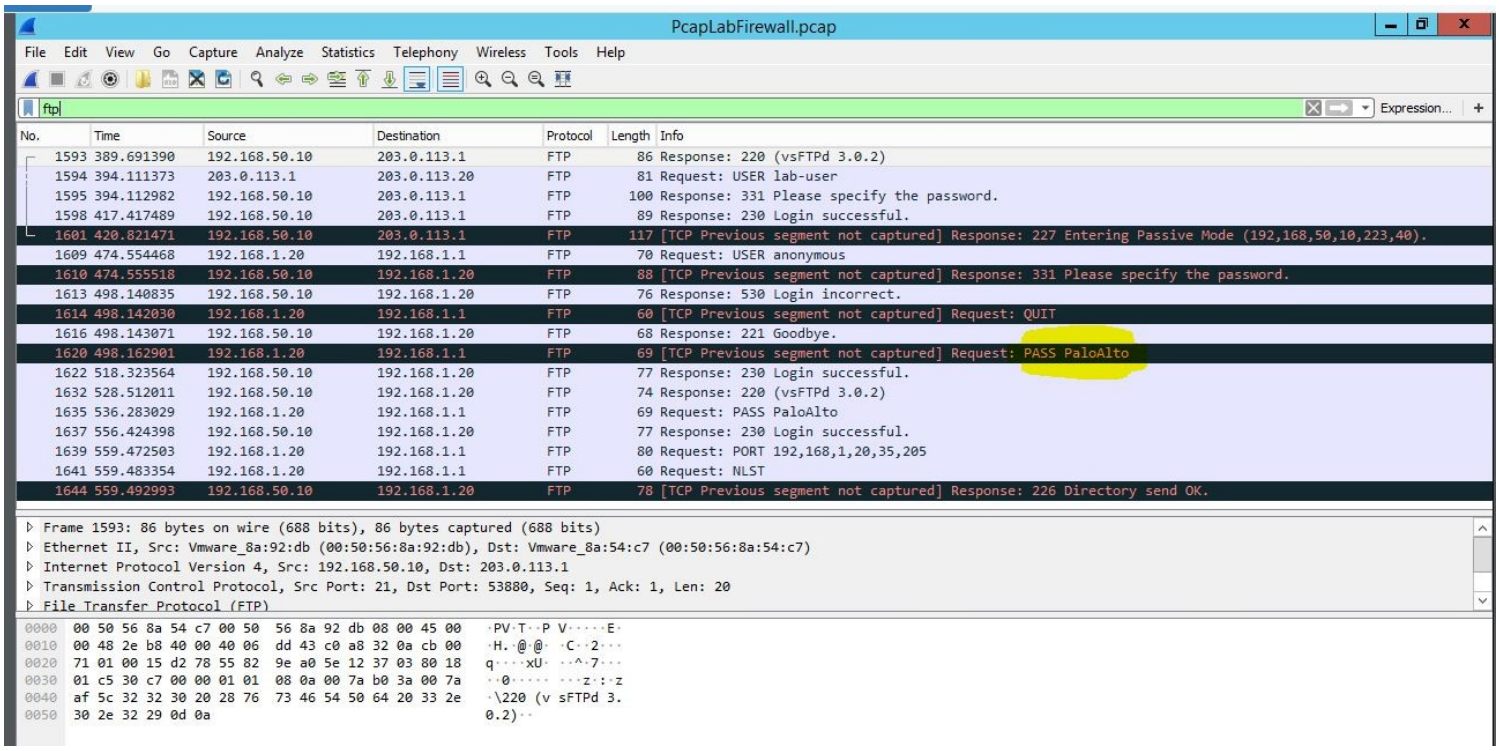
==========

Hello thank you for your time. Have you ever wondered about messages and sending them securely, by radio, email, and mail?

==========

Below is a photo of what happens with FTP (File Transfer Protocol) when used with a password. You can clearly see the password highlighted "PASS: PaloAlto" if you used a packet sniffer. PaloAlto is the pasword that I used in the cyber security IT140 lab and it is clearly visable inside of Wireshark. We would normally want to encode that Password and use something like SSH Secure Shell with a cipher or algorithm so no one can know what the password is.

==========



==========

Photo: Wireshark showing the password being sent to the system over clear text was intercepted with WireShark.

==========

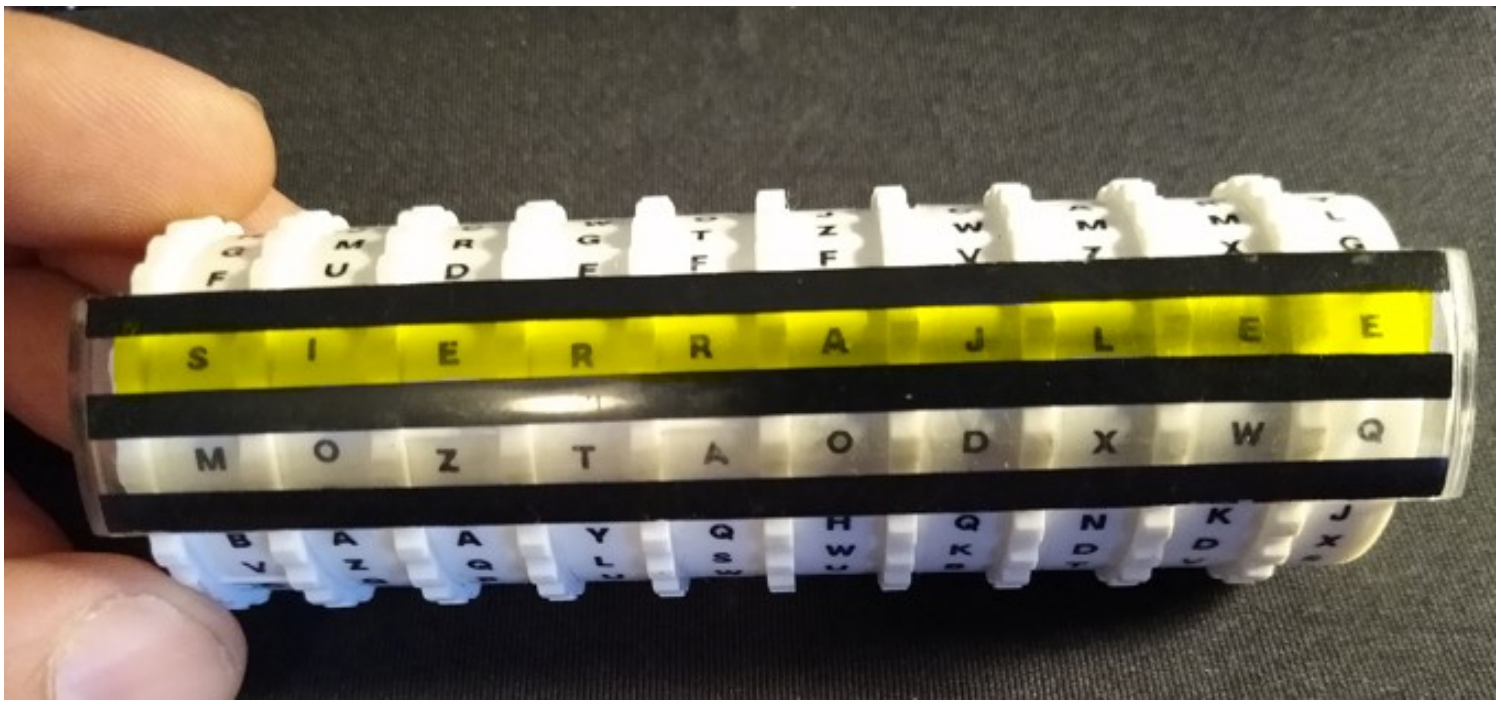
I hope to talk to you about cipher wheels. Mainly the "M-94 used durring 1917-1942." per http://www.jproc.ca/

==========

Lets start with some background on Cipher Wheels.

==========

==========
I have two jefferson cipher wheels I want to send a secure message, however I must first write my message. I want the message to say Sierra.
==========

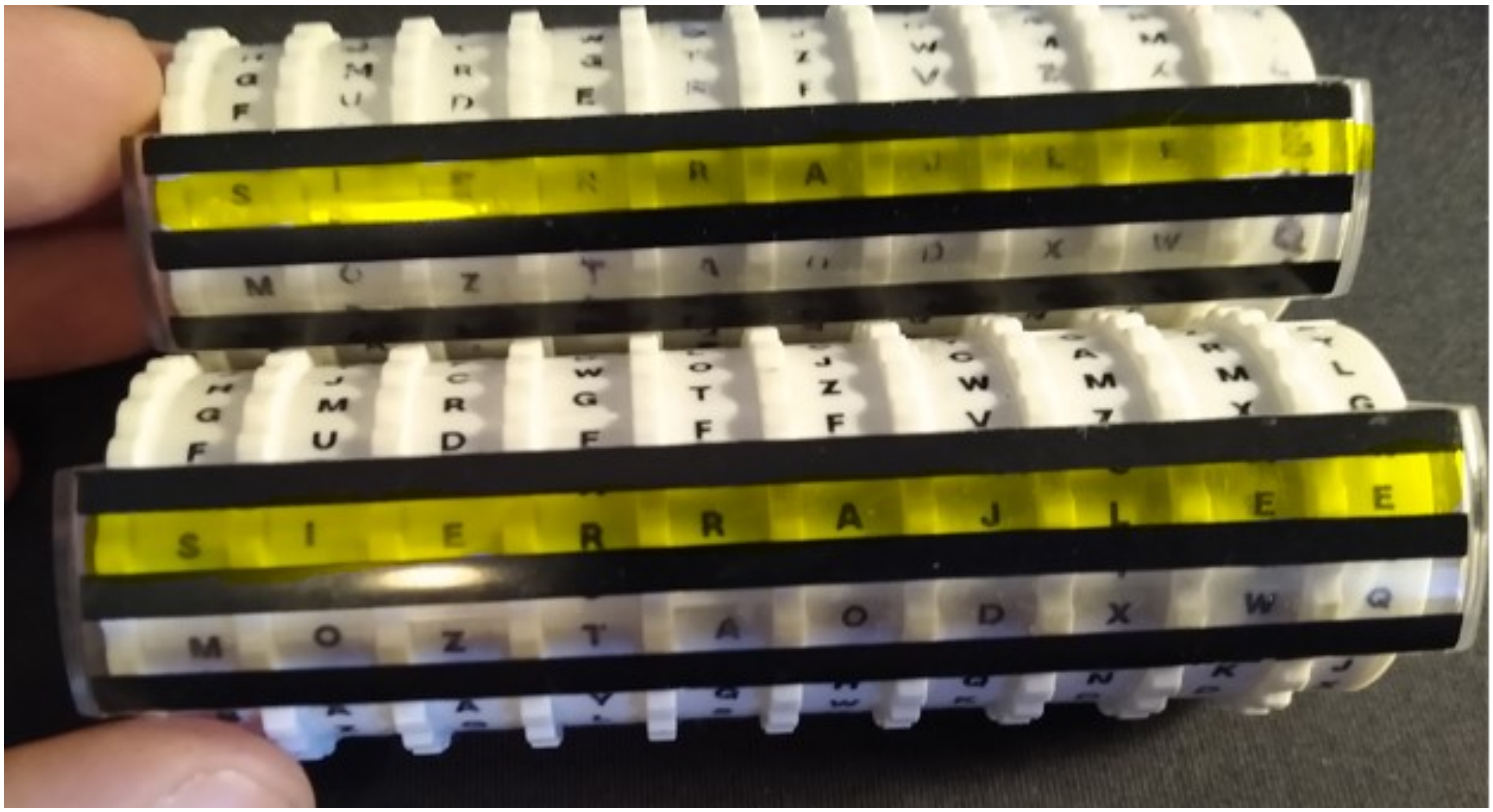


==========
So I set each disc to SIERRAJLEE and count the rows up or down in a range of 25. I used -9 so I count down 9 rows and find my cipher message that is PWISKBRFQC. This message is no longer in clear text it has been "ciphered or encoded." To send this coded message we send PWISKBRFQC over the messge medium with a number being “9.” Without the full algorithm or some program it is harder to understand. You would have to understand how this mesage was encoded. My jefferson cipher wheels have 26 letters and 10 wheels. This is very limited cryptographic security, however it is better than clear text for a password.
==========

==========

After I counted down 9 rows this is my encoded message PWISKBRFQC to send to the reciver.

==========



==========

On my reciever cipher wheel I place in the code PWISKBRFQC and count up 9 rows, revealing that secure message "SIERRAJLEE."

==========

==========

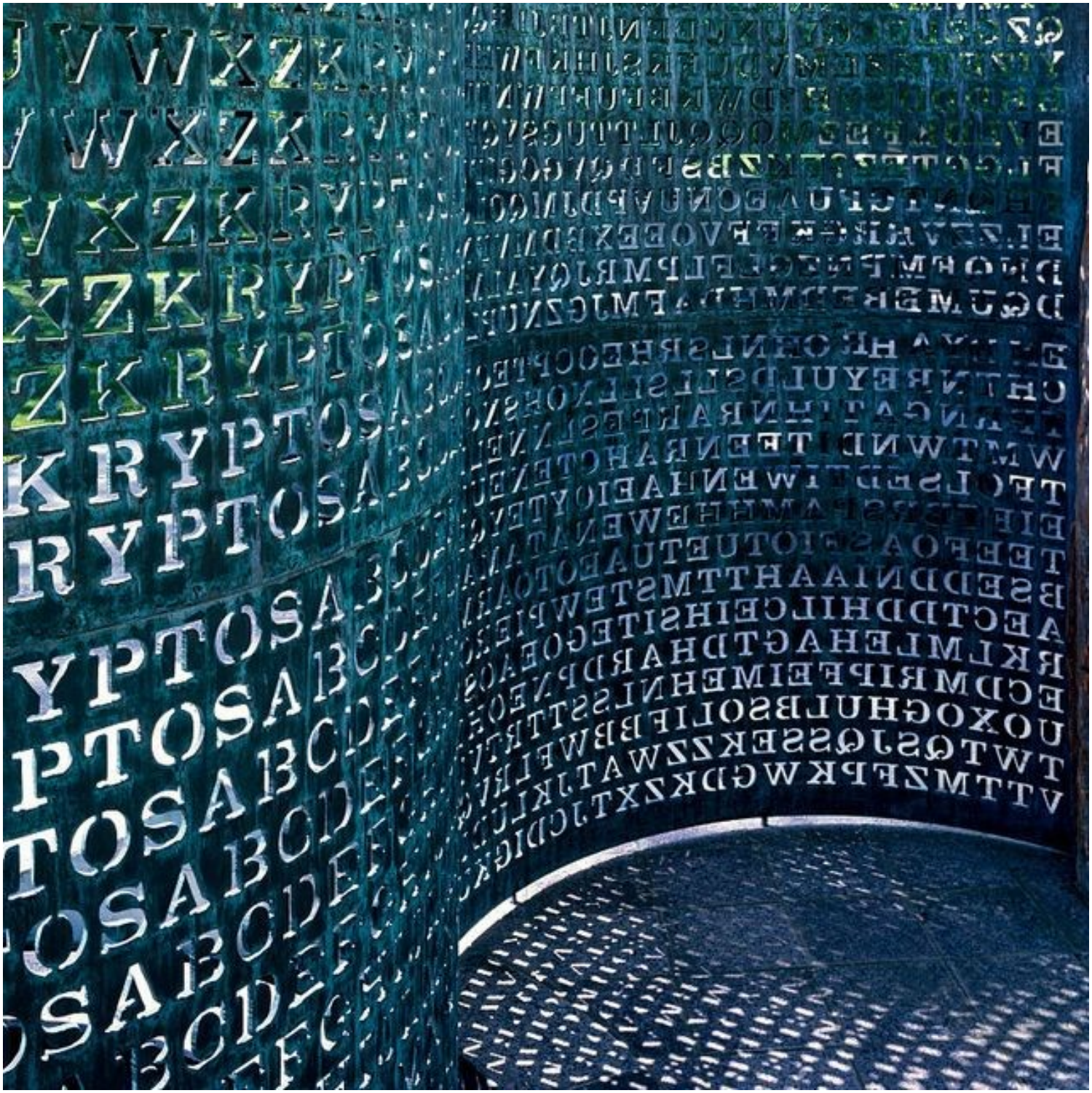
If you are old enough to remember on "A Christmas Story" the secret decoder pin that reveals "Drink more ovaltine" cipher wheels work on the same concept.

==========

==========

From the movie the secret decoder pin is seen above. Every letter went up or down a specific number of times. Picture a device with many wheels or discs all going up and down a specific number of times with different variations of the alphabet for each disc. This is what Kyptos is based on some algorithm of movements of the key to make the message.

==========



==========

"The 12-foot-high, verdigrised copper, granite and wood sculpture on the grounds of the CIA complex in Langley, Virginia, contains four encrypted messages carved out of the metal, three of which were solved years ago. The fourth is composed of just 97 letters, but its brevity belies its strength. Even the NSA, whose master crackers were the first to decipher other parts of the work, gave up on cracking it long ago. So four years ago, concerned that he might not live to see the mystery of Kryptos resolved, Sanborn released a clue to help things along, revealing that six of the last 97 letters when decrypted spell the word "Berlin"---a revelation that many took to be a reference to the Berlin Wall" (Zetter).

==========

==========

USA M-94 Information Cited from:
M-94, www.jproc.ca/crypto/m94.htm

==========

==========

"The M-94 was a W.W.II United States Army Signal Corps cipher device used from the early 1920's up till 1942 as a low level, tactical, cryptographic encoding/decoding device. A version used by the U.S. Navy was called CSP-488. It was also called a Jefferson device and was similar in type (if not identical) to Major Bazeries Cipher Device or Bazeries Cylinder. Over time, these units were replaced by the M-209 cipher machine.
Per Louis Kruh, about 9,432 of the M-94's were made between their adoption in 1921 and their replacement by the M-209 on August 9, 1943 . He notes about 150,000 of the M-209's were made" (jproc)."
Because so many M-94s were made I wanted to take Professor Gita's "Sierra College CSCI12 class Ceaser Cipher Assignment Program" and adapt it to work with more than one range of letters. I first set this up to mimic a digital version of the M-94 and added only Kyptos based keys. I serached for known solutions that that matched the sequence in my keys to help try to find any way to decode Kyptos.

==========

```
{
    String B1  = " ABCDEFGHIJKLMNOPQRSTUVWXYZABCD";
    String C2  = "AKRYPTOSABCDEFGHIJLMNQUVWXZKRYP";
    String D3  = "BRYPTOSABCDEFGHIJLMNQUVWXZKRYPT";
    String E4  = "CYPTOSABCDFFGHIJLMNQUVWXZKRYPTO";
    String F5  = "DPTOSABCDEFGHIJLMNQUVWXZKRYPTOS";
    String G6  = "ETOSABCDEFGHIJLMNQUVWXZKRYPTOSA";
    String H7  = "FOSABCDEFGHIJLMNQUVWXZKRYPTOSAB";
    String I8  = "GSABCDEFGHIJLMNQUVWXZKRYPTOSABC";
    String J9 =  "HABCDEFGHIJLMNQUVWXZKRYPTOSABCD";
    String K10 = "IBCDEFGHIJLMNQUVWXZKRTPTOSABCDE";
    String L11 = "JCDEFGHIJLMNQUVWXZKRYPTOSABCDEF";
    String M12 = "KDEFGHIJLMNQUVWXZKRYPTOSABCDEFG";
    String N13 = "LEFGHIJKMNQUVWXZKRYPTOSABCDEFGH";
    String O14 = "MFGHIJLMNQUVWXZKRYPTOSABCDEFGHI";
    String P15 = "NGHIJLMNQUVWXZKRYPTOSABCDEFGHIJL";
    String Q16 = "OHIJLMNQUVWXZKRYPTOSABCDEFGHIJL";
    String R17 = "PIJLMNQUVWXZKRYPTOSABCDEFGHIJLM";
    String S18 = "QJLMNQUVWXZKRYPTOSABCDEFGHIJLMN";
    String T19 = "RLMNQUVWXZKRYPTOSABCDEFGHIJLMNQ";
    String U20 = "SMNQUVWXZKRYPTOSABCDEFGHIJLMNQU";
    String V21 = "TNQUVWXZKRYPTOSABCDEFGHIJLMNQUV";
    String W22 = "UQUVWXZKRYPTOSABCDEFGHIJLMNQUVW";
    String X23 = "VUVWXZKRYPTOSABCDEFGHIJLMNQUVWX";
    String Y24 = "WVWXZKRYPTOSABCDEFGHIJLMNQUVWXZ";
    String Z25 = "XWXZKRYPTOSABCDEFGHIJLMNQUVWXZK";
    String a26 = "YXZKRYPTOSABCDEFGHIJLMNQUVWXZKR";
    String b27 = "ZZKRYPTOSABCDEFGHIJLMNQUVWXZKRY";
    String c28 = " ABCDEFGHIJKLMNOPQRSTUVWXYZABCD";
```

==========

First let's set each wheel for a value similar to that of the Kyptos sculpture it self seen below as "THE KEY." Our wheels are digital and will act like the wheel seen above in the photo, the offset acts as what number of times we turn each wheel to after we have found our matched letter. A will turn up 3 times when set to offset of 3 for B1 and land at D as we count from zero up. A on C2 will end up on P, A on D3 will end up on

When we cipher out text we take each letter in ABCDEFGHIJKLMNOPQRTUVWXYZ and it will be ciphered to the offset of our chose we can use -numbers or postive numbers.

==========

==========

This key uses 28 wheels matched with my our program.

Levy, S. (2009, April 20). *Mission impossible: The code even the CIA CAN'T CRACK*. Wired. Retrieved October 1, 2021, from https://www.wired.com/2009/04/ff-kryptos/.

==========

```
String[] wheelarray = {B1, C2, D3, E4, F5, G6, H7, I8, J9, K10, L11, M12, N13, O14, P15, Q16, R17, S18, T19, U20, V21, W22, X23, Y24, Z25, a26, b27, c28};

//System.out.println(wheelarray[2]); // testing for array print per numerical use
Scanner kb = new Scanner(System.in);
System.out.print("Enter your offset for M94 cipher 1-31 or negative offsets: ");
while(!kb.hasNextInt())
{
   kb.next(); // discard input
   System.out.println("Error! Please only enter a numerical offset cipher discs");
   System.out.print("Enter your offset for M94 cipher: ");
}
int offset = kb.nextInt();
kb.nextLine();//flush buffer
System.out.print("Enter a message to encrypt with a M94 cipher 31 letter max per input: ");
String testing = kb.nextLine();
testing = testing.toUpperCase();
int len = testing.length();
int i = 0;
int j = 0;
int s = 0;
String array_in_use = B1;
while (i<len) // i is less then the length of hello
{
   char letter = testing.charAt(i);
   {
      s++;
   }
   s = s + offset;
   if (s>30)
```

==========

Next lets make our array based on a an array of strings to iterate over. We have some data validation and requirements for setting the encryption up.

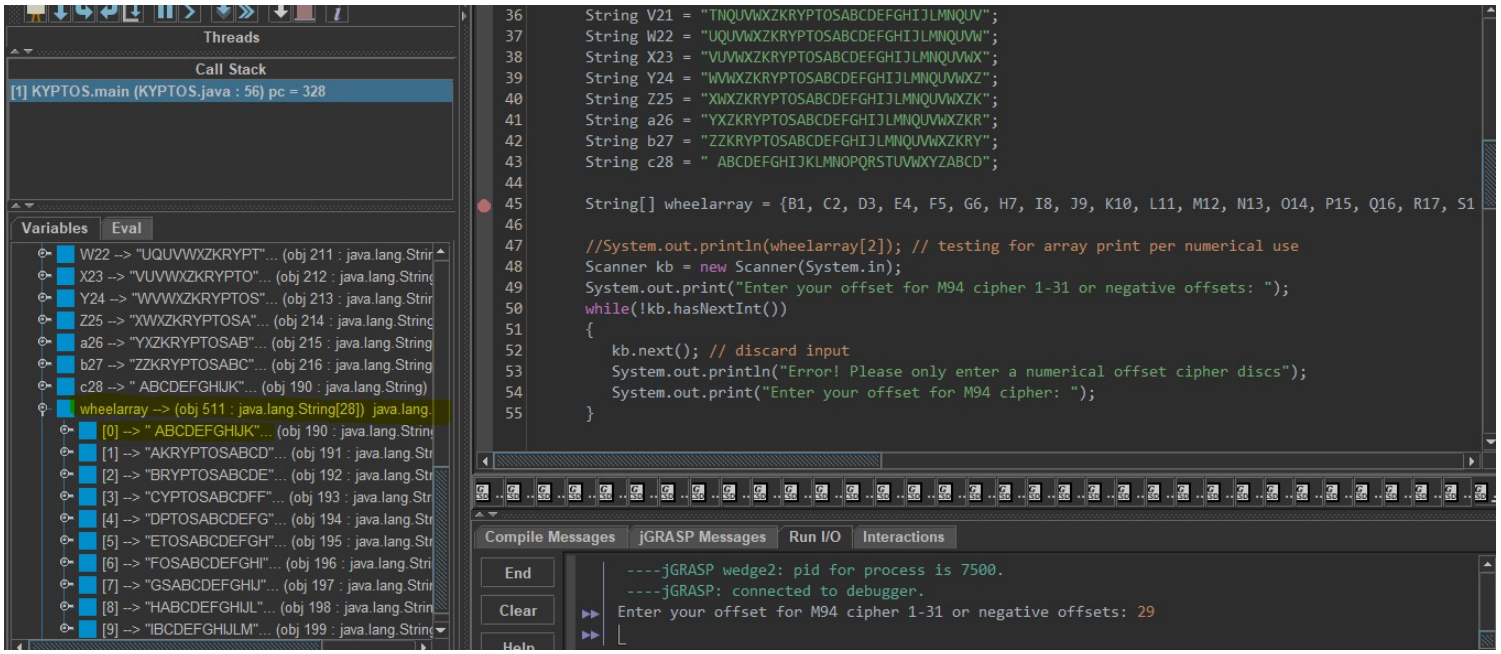
==========

```
Enter your offset for M94 cipher 1-31 or negative offsets:
29
Enter a message to encrypt with a M94 cipher 31 letter max per input: abcdefghijklmnopqrstuvwxyz
DPTOSABCDEFGHIJLMNQUVWXZKR
```

==========
Notice we used a offset of 29 for this and used the the alaphebet.
This gave us the last wheel letter for each disc. Our offset does not vary for each character this is for basic undersanding.
When we set our offsets it will find the letter and count our disc up and down to match each location.
31 letters are on each wheel for Kytpos, and the ability to offset negatives or postives. Each letters location counts up or down any number of times.
==========

```
String B1  = " ABCDEFGHIJKLMNOPQRSTUVWXYZABCD";
String C2  = "AKRYPTOSABCDEFGHIJLMNQUVWXZKRYP";
String D3  = "BRYPTOSABCDEFGHIJLMNQUVWXZKRYPT";
String E4  = "CYPTOSABCDFFGHIJLMNQUVWXZKRYPTO";
String F5  = "DPTOSABCDEFGHIJLMNQUVWXZKRYPTOS";
String G6  = "ETOSABCDEFGHIJLMNQUVWXZKRYPTOSA";
String H7  = "FOSABCDEFGHIJLMNQUVWXZKRYPTOSAB";
String I8  = "GSABCDEFGHIJLMNQUVWXZKRYPTOSABC";
String J9 =  "HABCDEFGHIJLMNQUVWXZKRYPTOSABCD";
String K10 = "IBCDEFGHIJLMNQUVWXZKRTPTOSABCDE";
String L11 = "JCDEFGHIJLMNQUVWXZKRYPTOSABCDEF";
String M12 = "KDEFGHIJLMNQUVWXZKRYPTOSABCDEFG";
String N13 = "LEFGHIJKMNQUVWXZKRYPTOSABCDEFGH";
String O14 = "MFGHIJLMNQUVWXZKRYPTOSABCDEFGHI";
String P15 = "NGHIJLMNQUVWXZKRYPTOSABCDEFGHIJL";
String Q16 = "OHIJLMNQUVWXZKRYPTOSABCDEFGHIJL";
String R17 = "PIJLMNQUVWXZKRYPTOSABCDEFGHIJLM";
String S18 = "QJLMNQUVWXZKRYPTOSABCDEFGHIJLMN";
String T19 = "RLMNQUVWXZKRYPTOSABCDEFGHIJLMNQ";
String U20 = "SMNQUVWXZKRYPTOSABCDEFGHIJLMNQU";
String V21 = "TNQUVWXZKRYPTOSABCDEFGHIJLMNQUV";
String W22 = "UQUVWXZKRYPTOSABCDEFGHIJLMNQUVW";
String X23 = "VUVWXZKRYPTOSABCDEFGHIJLMNQUVWX";
String Y24 = "WVWXZKRYPTOSABCDEFGHIJLMNQUVWXZ";
String Z25 = "XWXZKRYPTOSABCDEFGHIJLMNQUVWXZK";
String a26 = "YXZKRYPTOSABCDEFGHIJLMNQUVWXZKR";
String b27 = "ZZKRYPTOSABCDEFGHIJLMNQUVWXZKRY";
String c28 = " ABCDEFGHIJKLMNOPQRSTUVWXYZABCD";
```

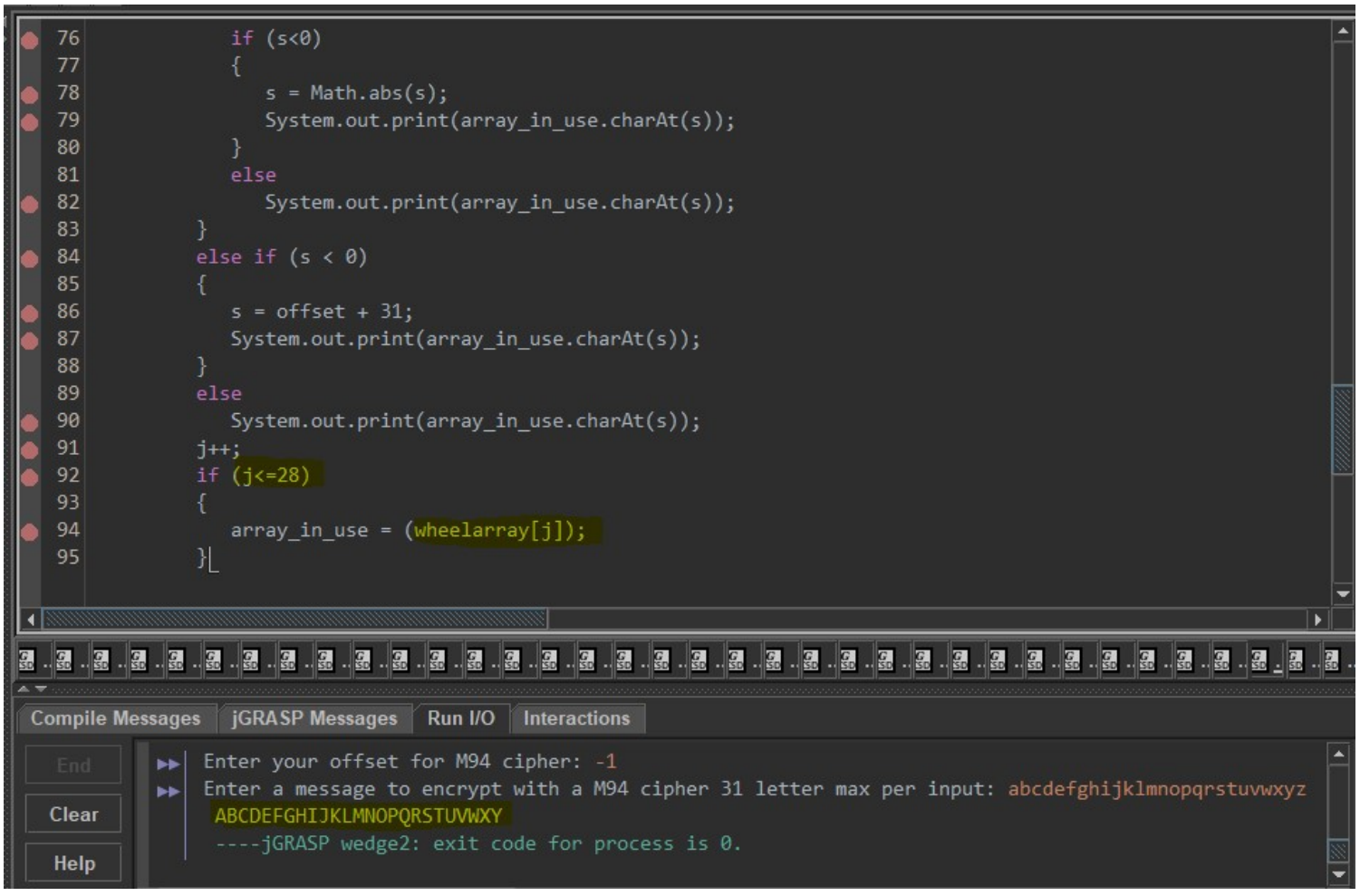
==========
This has 28 discs versus the M-94 at only 25. Thats what causes some consusion. Each wheel has 31 characters.
==========



==========

Our disc array now is set up to include every disc in the key above. The offset will roll the digial disc to the character in that location for each letter we used.

==========



==========

This will iterate over each wheel from 1-28 and will offset the char or character you have in place for that location. This is a digital version of a M-94 chiper that is adapted for use with only Kyptos keys. It is a start of a encryption program. If you want to learn more be sure to enroll in CSCI12 at Sierra College.

==========

==========
Please understand that this was worked on with trial and error. I had to rework my original M-94 program to add real locations because they are no longer in ABC sequential order as the numbers exceed 26 letters. For this I added an adaption to the chiper program. This now will first find the location of each letter within each wheel and after roatate them to that and stop. Seen here.
==========

```
    while(i<len) // i is less then the length of hello
    {
      String array_in_use = (wheelarray[i]);
      char letter = testing.charAt(i);
      j = array_in_use.indexOf(testing.charAt(i));
      s = j + offset;
      if (s>30)
      {
        s = 31-offset;
        if (s<0)
        {
          s = Math.abs(s);
          System.out.print(array_in_use.charAt(s));
        }
        else
          System.out.print(array_in_use.charAt(s));
      }
      else if (s < 0)
      {
        s = offset + 31;
        System.out.print(array_in_use.charAt(s));
      }
      else
        System.out.print(array_in_use.charAt(s));
      i++;
      s=0;
    }
  }
}
```

==========
I hope now you can understand what it takes to decode messages that are sent over shortwave radio durring WW2. My first program failed with the correct rolls. It should first make a mesages and rotate each wheel a given number of times via the offset and add know what duplicate letter is used. This will give a new letter that is set as the coded letter.
==========

```
12    public static void main (String args[])
13    {
14       String B1  = " ABCDEFGHIJKLMNOPQRSTUVWXYZABCD";
15       String C2  = "AKRYPTOSABCDEFGHIJLMNQUVWXZKRYP";
16       String D3  = "BRYPTOSABCDEFGHIJLMNQUVWXZKRYPT";
17       String E4  = "CYPTOSABCDFFGHIJLMNQUVWXZKRYPTO";
```

```
Compile Messages | jGRASP Messages | Run I/O | Interactions

End
Clear
Help

Enter your offset for cipher 1-31 or negative offsets: 3
Enter a message to encrypt with a M94 cipher 31 letter max per input: SIERRACOLLEGE
VMHTTDFBQQHJH
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java KYPTOS
 ----   at: Oct 3, 2021, 1:38:18 PM

 ----jGRASP wedge: pid for wedge is 6016.
 ----jGRASP wedge2: pid for wedge2 is 2380.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Documents].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" KYPTOS].
 ----jGRASP wedge2: pid for process is 6472.
Enter your offset for cipher 1-31 or negative offsets: -3
Enter a message to encrypt with a M94 cipher 31 letter max per input: VMHTTDFBQQHJH
SIECTASGLLEGE

Line:90  Col:5  Code:0  Top:12
```

==========

What makes Kryptos so hard is that each cipher line contains dual letters that relate to Kryptos. For example line 3 contains two R's so the code has to be adapted for a different algorithm.

==========

```
Enter your offset for M94 cipher 1-25 or negative offsets: 5
Enter a message to encrypt with a M94 cipher 25 letter max per input: HelloSierraCollege
OKMOBMOUKUXYARPYLP
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java JonathanLeeM94CipherProgram2021PeerTutor
 ----   at: Oct 3, 2021, 9:51:48 PM

 ----jGRASP wedge: pid for wedge is 6736.
 ----jGRASP wedge2: pid for wedge2 is 7136.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Desktop\Peer Tutor].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" Jonathan
 ----jGRASP wedge2: pid for process is 11048.
Enter your offset for M94 cipher 1-25 or negative offsets: -5
Enter a message to encrypt with a M94 cipher 25 letter max per input: OKMOBMOUKUXYARPYLP
HELLOSIERRACOLLEGE
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.
```

==========

My M-94 cipher program works perfectly. However Kryptos has multiple letters in each string causing issues with finding a algorithm that works to decode and encode a given message.

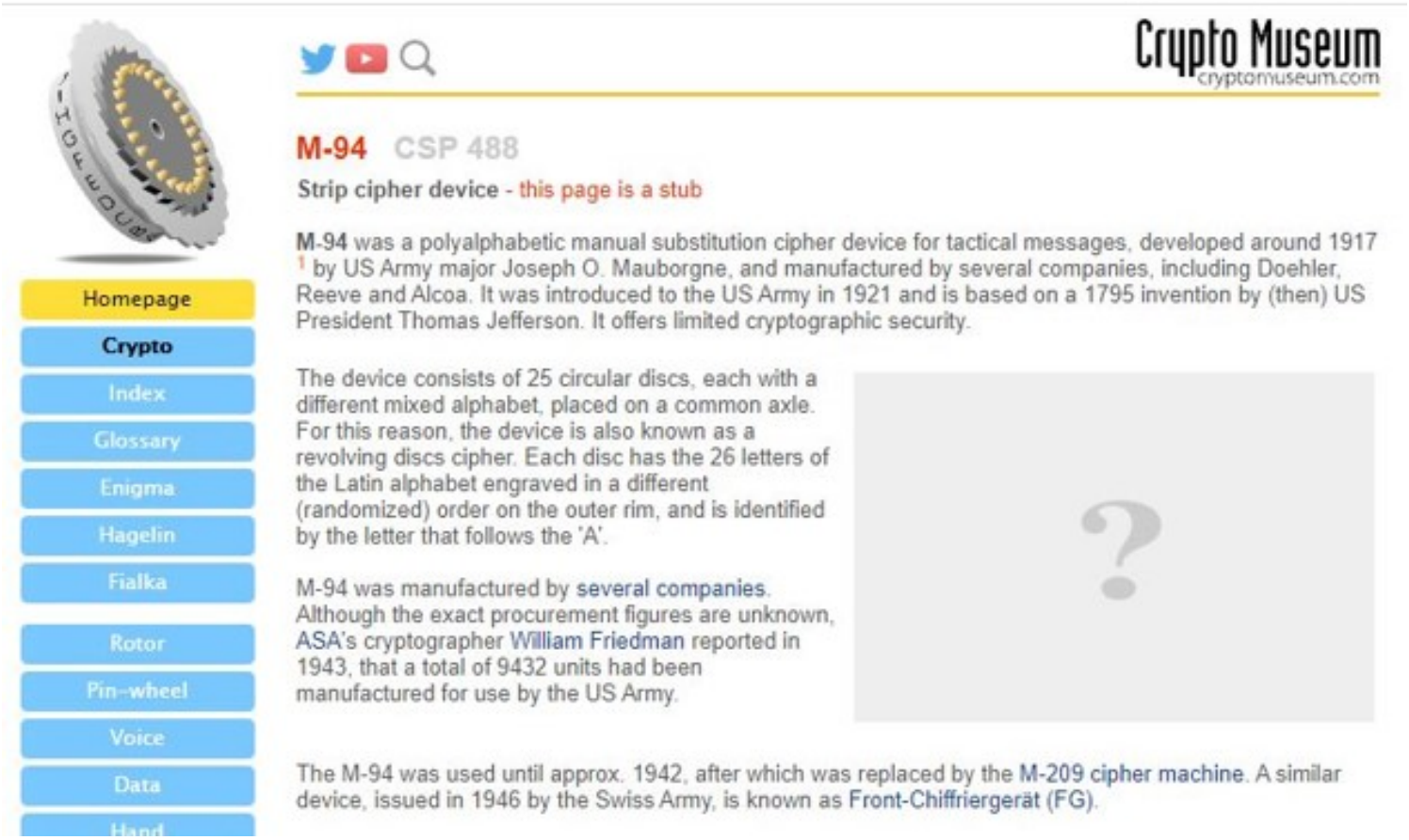
==========

```
Enter your offset for M94 cipher 1-25 or negative offsets: 5
Enter a message to encrypt with a M94 cipher 25 letter max per input: Hello Sierra College
OKMOB CBUUAH ERDMPMW
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java JonathanLeeM94CipherProgram2021PeerTutor
 ----   at: Oct 3, 2021, 9:53:28 PM

 ----jGRASP wedge: pid for wedge is 9848.
 ----jGRASP wedge2: pid for wedge2 is 11156.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Desktop\Peer Tutor].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" Jonathan
 ----jGRASP wedge2: pid for process is 4960.
Enter your offset for M94 cipher 1-25 or negative offsets: -5
Enter a message to encrypt with a M94 cipher 25 letter max per input: OKMOB CBUUAH ERDMPMW
HELLO SIERRA COLLEGE
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.
```

==========

The M-94 was easier to work with as it only uses 26 letters per disc and each letter is different. According to cyptomuseum.com *"The device consists of 25 circular discs, each with a different mixed alphabet, placed on a common axle. For this reason, the device is also known as a revolving discs cipher."*

==========



Crypto Museum
cryptomuseum.com

Homepage
Crypto
Index
Glossary
Enigma
Hagelin
Fialka
Rotor
Pin-wheel
Voice
Data

M-94 CSP 488

Strip cipher device - this page is a stub

M-94 was a polyalphabetic manual substitution cipher device for tactical messages, developed around 1917 [1] by US Army major Joseph O. Mauborgne, and manufactured by several companies, including Doehler, Reeve and Alcoa. It was introduced to the US Army in 1921 and is based on a 1795 invention by (then) US President Thomas Jefferson. It offers limited cryptographic security.

The device consists of 25 circular discs, each with a different mixed alphabet, placed on a common axle. For this reason, the device is also known as a revolving discs cipher. Each disc has the 26 letters of the Latin alphabet engraved in a different (randomized) order on the outer rim, and is identified by the letter that follows the 'A'.

M-94 was manufactured by several companies. Although the exact procurement figures are unknown, ASA's cryptographer William Friedman reported in 1943, that a total of 9432 units had been manufactured for use by the US Army.

The M-94 was used until approx. 1942, after which was replaced by the M-209 cipher machine. A similar device, issued in 1946 by the Swiss Army, is known as Front-Chiffriergerät (FG).

==========

Photo of website "The Crypto Museum"

==========

```
{
    String B1  = "ABCEIGDJFVUYMHTQKZOLRXSPWN";
    String C2  = "ACDEHFIJKTLMOUVYGZNPQXRWSB";
    String D3  = "ADKOMJUBGEPHSCZINXFYQRTVWL";
    String E4  = "AEDCBIFGJHLKMRUOQVPTNWYXZS";
    String F5  = "AFNQUKDOPITJBRHCYSLWEMZVXG";
    String G6  = "AGPOCIXLURNDYZHWBJSQFKVMET";
    String H7  = "AHXJEZBNIKPVROGSYDULCFMQTW";
    String I8  = "AIHPJOBWKCVFZLQERYNSUMGTDX";
    String J9  = "AJDSKQOIVTZEFHGYUNLPMBXWCR";
    String K10 = "AKELBDFJGHONMTPRQSVZUXYWIC";
    String L11 = "ALTMSXVQPNOHUWDIZYCGKRFBEJ";
    String M12 = "AMNFLHQGCUJTBYPZKXISRDVEWO";
    String N13 = "ANCJILDHBMKGXUZTSWQYVORPFE";
    String O14 = "AODWPKJVIUQHZCTXBLEGNYRSMF";
    String P15 = "APBVHIYKSGUENTCXOWFQDRLJZM";
    String Q16 = "AQJNUBTGIMWZRVLXCSHDEOKFPY";
    String R17 = "ARMYOFTHEUSZJXDPCWGQIBKLNV";
    String S18 = "ASDMCNEQBOZPLGVJRKYTFUIWXH";
    String T19 = "ATOJYLFXNGWHVCMIRBSEKUPDZQ";
    String U20 = "AUTRZXQLYIOVBPESNHJWMDGFCK";
    String V21 = "AVNKHRGOXEYBFSJMUDQCLZWTIP";
    String W22 = "AWVSFDLIEBHKNRJQZGMXPUCOTY";
    String X23 = "AXKWREVDTUFOYHMLSIQNJCPGBZ";
    String Y24 = "AYJPXMVKBQWUGLOSTECHNZFRID";
    String Z25 = "AZDNBUHYFWJLVGRCQMPSOEXTKI";
```

==========

M-94 Ciper Wheels added to program. On Disc R17 it shows ARMYOFTHEUS
Wheel Information provided by "http://www.jproc.ca/crypto/m94.html"
Please note we are using a standard disc pattern of B1 to Z25 the discs can be placed and adapted to any pattern needed.

The real M-94 can have all the discs changed to any pattern also.

==========

```
Enter your offset for M94 cipher 1-25 or negative offsets: 25
Enter a message to encrypt with a M94 cipher 25 letter max per input: aaaaaaaaaaaaaaaaaaaaaaaaa
NBLSGTWXRCJOEFMYVHQKPYZDI
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java JonathanLeeM94CipherProgram2021PeerTutor
 ----   at: Oct 3, 2021, 9:56:42 PM

 ----jGRASP wedge: pid for wedge is 2792.
 ----jGRASP wedge2: pid for wedge2 is 11236.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Desktop\Peer Tutor].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" Jonathan
 ----jGRASP wedge2: pid for process is 3144.
Enter your offset for M94 cipher 1-25 or negative offsets: -25
Enter a message to encrypt with a M94 cipher 25 letter max per input: NBLSGTWXRCJOEFMYVHQKPYZDI
AAAAAAAAAAAAAAAAAAAAAAAAA
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.
```

==========

Notice all A's will take us to the last line when 25 offset is used.

==========

```
Enter your offset for M94 cipher 1-25 or negative offsets: 5
Enter a message to encrypt with a M94 cipher 25 letter max per input: ooooooooooooooooooooooooo
PZGNBUUVERILAJRAUVXSFVSHI
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java JonathanLeeM94CipherProgram2021PeerTutor
 ----   at: Oct 3, 2021, 9:58:04 PM

 ----jGRASP wedge: pid for wedge is 1524.
 ----jGRASP wedge2: pid for wedge2 is 984.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Desktop\Peer Tutor].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" Jonathan
 ----jGRASP wedge2: pid for process is 3280.
Enter your offset for M94 cipher 1-25 or negative offsets: -5
Enter a message to encrypt with a M94 cipher 25 letter max per input: PZGNBUUVERILAJRAUVXSFVSHI
OOOOOOOOOOOOOOOOOOOOOOOOO
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.
```

==========

But let's test something more random, like "O" on each disc is set in a differnt location. On disc M12 the letter O is on the last location. So we tested with O and set it to offset 5

==========

```
Enter your offset for M94 cipher 1-25 or negative offsets: -5
Enter a message to encrypt with a M94 cipher 25 letter max per input: Thank you for your time
VBROA VAE YRW PCOP FCRA
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.

 ----jGRASP exec: java JonathanLeeM94CipherProgram2021PeerTutor
 ----   at: Oct 3, 2021, 10:00:12 PM

 ----jGRASP wedge: pid for wedge is 6632.
 ----jGRASP wedge2: pid for wedge2 is 248.
 ----jGRASP wedge2: CLASSPATH is ";.;;;C:\Program Files (x86)\jGRASP\extensions\classes".
 ----jGRASP wedge2: working directory is [C:\Users\jonat\OneDrive\Desktop\Peer Tutor].
 ----jGRASP wedge2: actual command sent ["C:\Program Files\Java\jdk-15.0.2\bin\java.EXE" Jonathan
 ----jGRASP wedge2: pid for process is 3572.
Enter your offset for M94 cipher 1-25 or negative offsets: 5
Enter a message to encrypt with a M94 cipher 25 letter max per input: VBROA VAE YRW PCOP FCRA
THANK YOU FOR YOUR TIME
 ----jGRASP wedge2: exit code for process is 0.
 ----jGRASP: operation complete.
```

==========

Thank you for your time.

==========

Multithreading

==========

This program is set up to have a user give it a number and it will break up that number into manageable parts to compute all perfect numbers inside of it. The first set is small and quick so we will use only 1/4 of the numbers, The higher numbers takes longer so we will break them into 8ths for the larger numbers to compute.
I worked with the thread class on page 975 of Daniel Liang's Introduction to Java Programing to help find a solution.

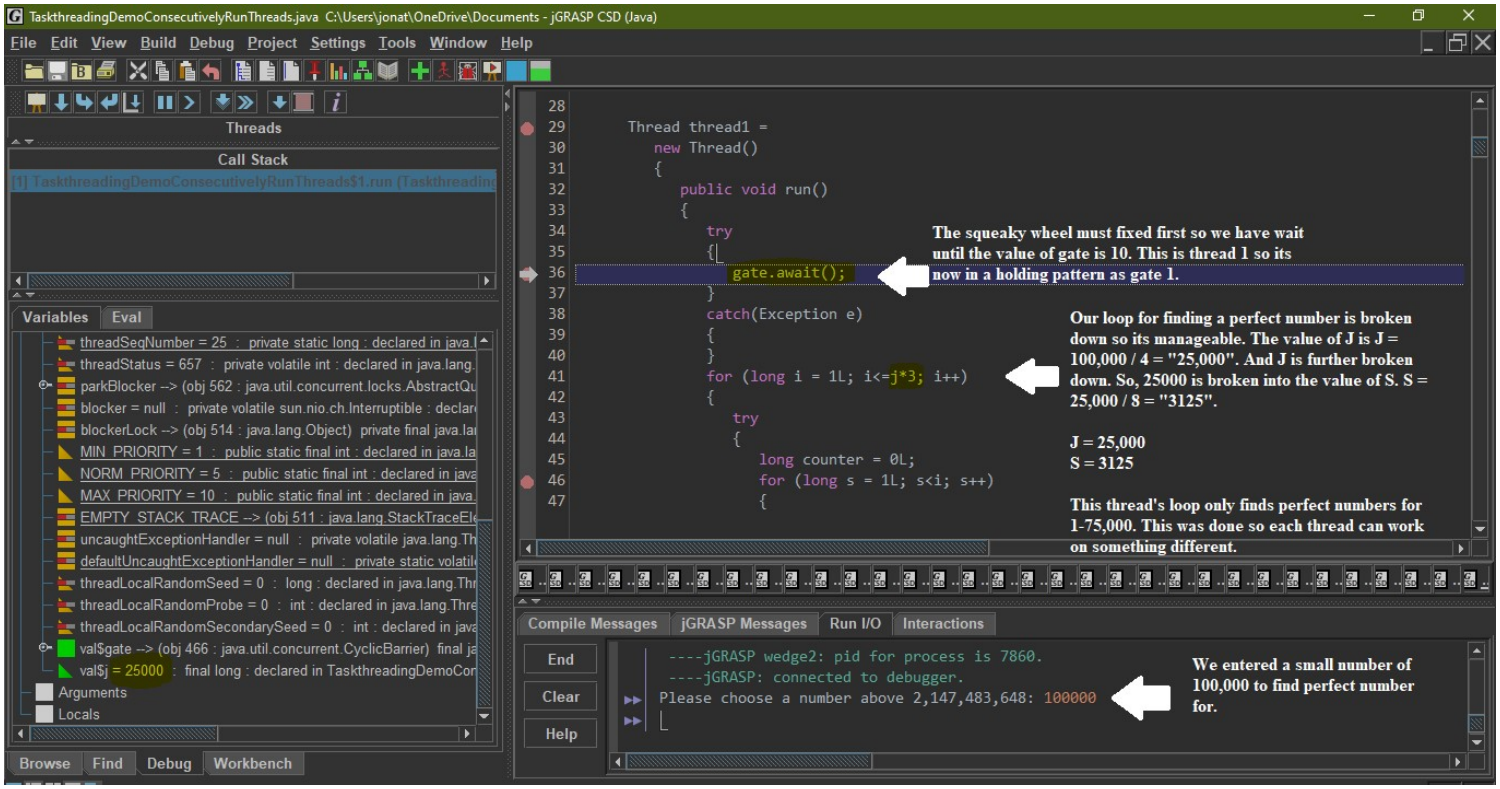
==========

```
15
16 import java.util.*;
17 import java.util.concurrent.CyclicBarrier;
18 public class TaskthreadingDemoConsecutivelyRunThreads
19 {
20    public static void main (String args[]) throws Exception
21    {
22       Scanner console = new Scanner(System.in);
23       System.out.print("Please choose a number above 2,147,483,648: ");
24       long numbercall = console.nextLong();
25       CyclicBarrier gate = new CyclicBarrier(10);
26       long j = numbercall/4; // this sets up the number into 4ths
27       long s = (numbercall-j*3)/8; // this takes the uper 25% of the main number and breaks it into 3rds
28
```

==========

So we have a total of 9 parts 1/4th of the numbers is in thread 1 and the rest is broken up into 8 other threads. Keep in mind this can be done a better way. This is for simple understanding of threads and gates.

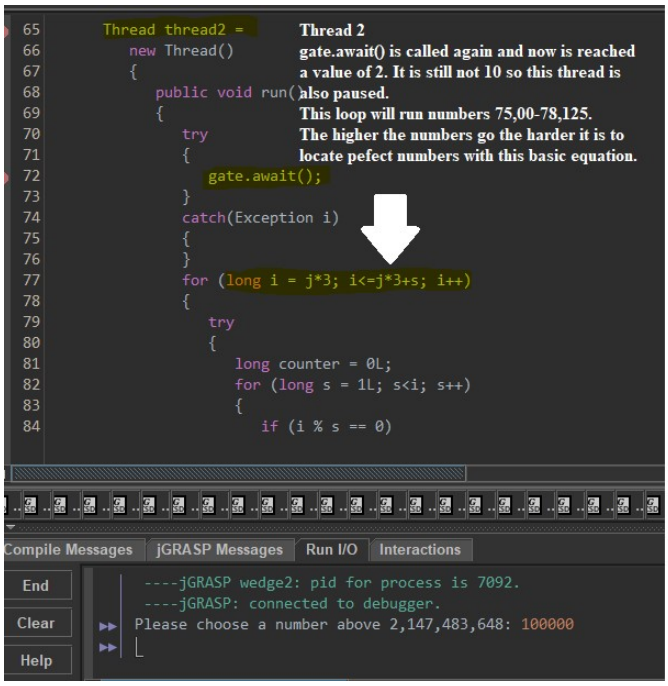
==========



==========

This is our first thread, 1-75,000. Java can can do many things at once we paused this thread see the gate.await() it has a value now of 1. Once it has been started.

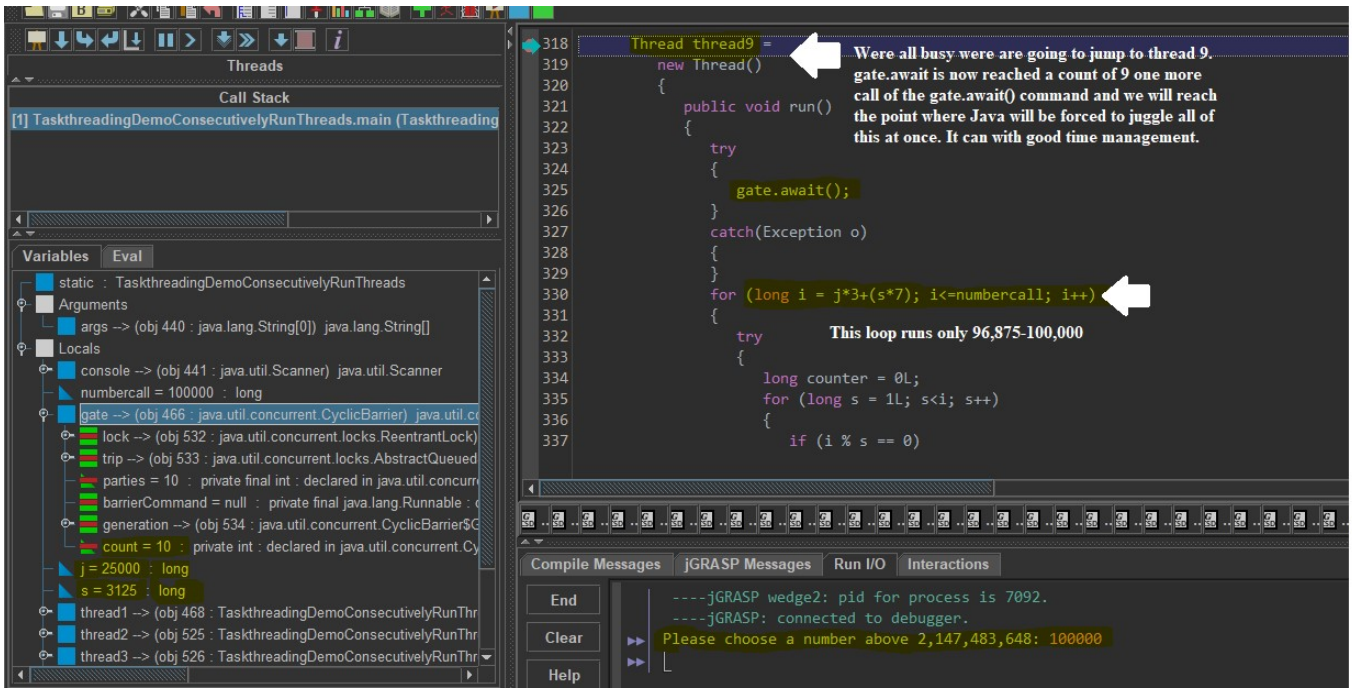
==========



==========

Thread 2 is for 75,000-78,125. Gate is a value of 2 so its on hold.

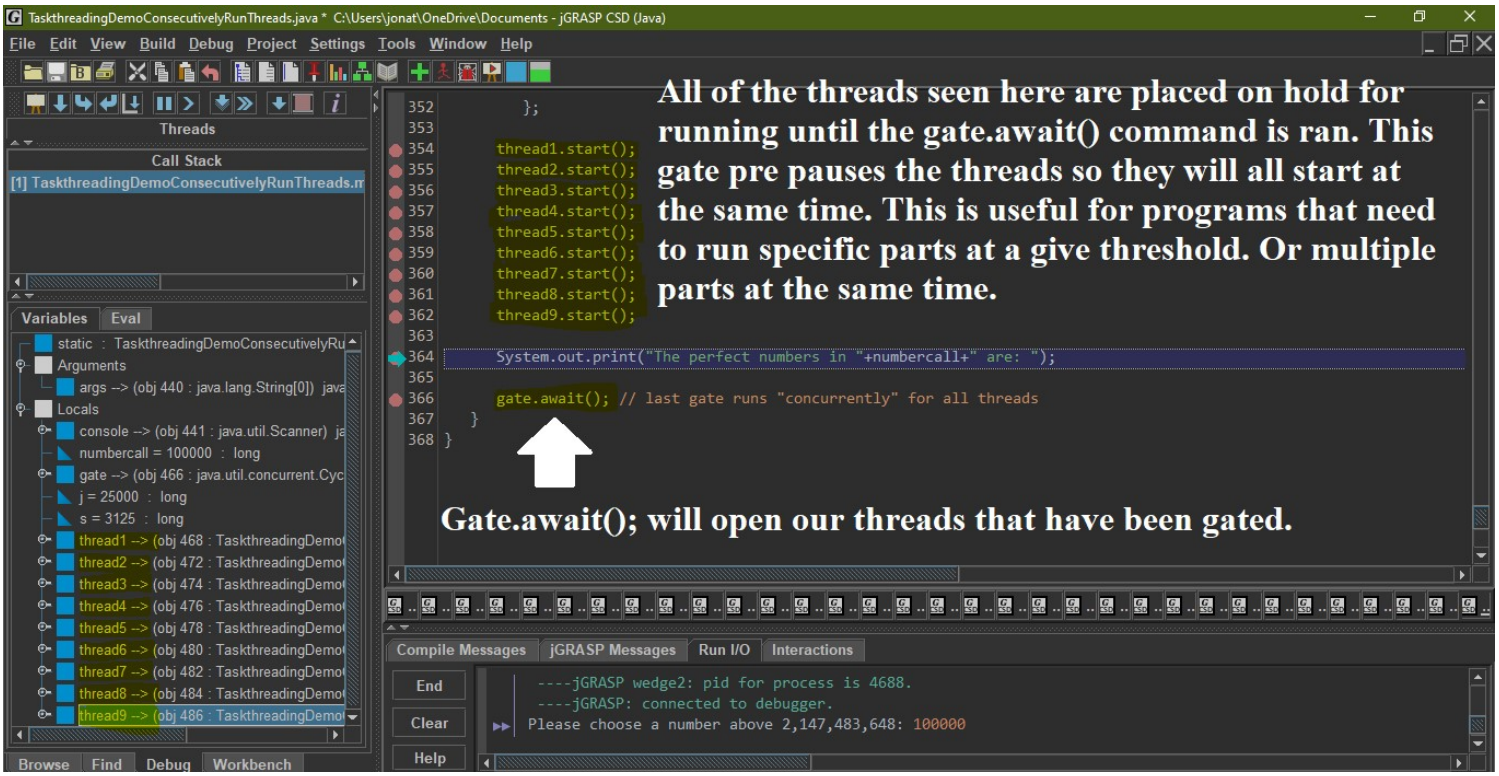
==========



==========

We jumped ahead to loop 9 this runs numbers from 96,875-100,000 and our gate is at 9. We need to call the gate one more time before it will run everything at once.

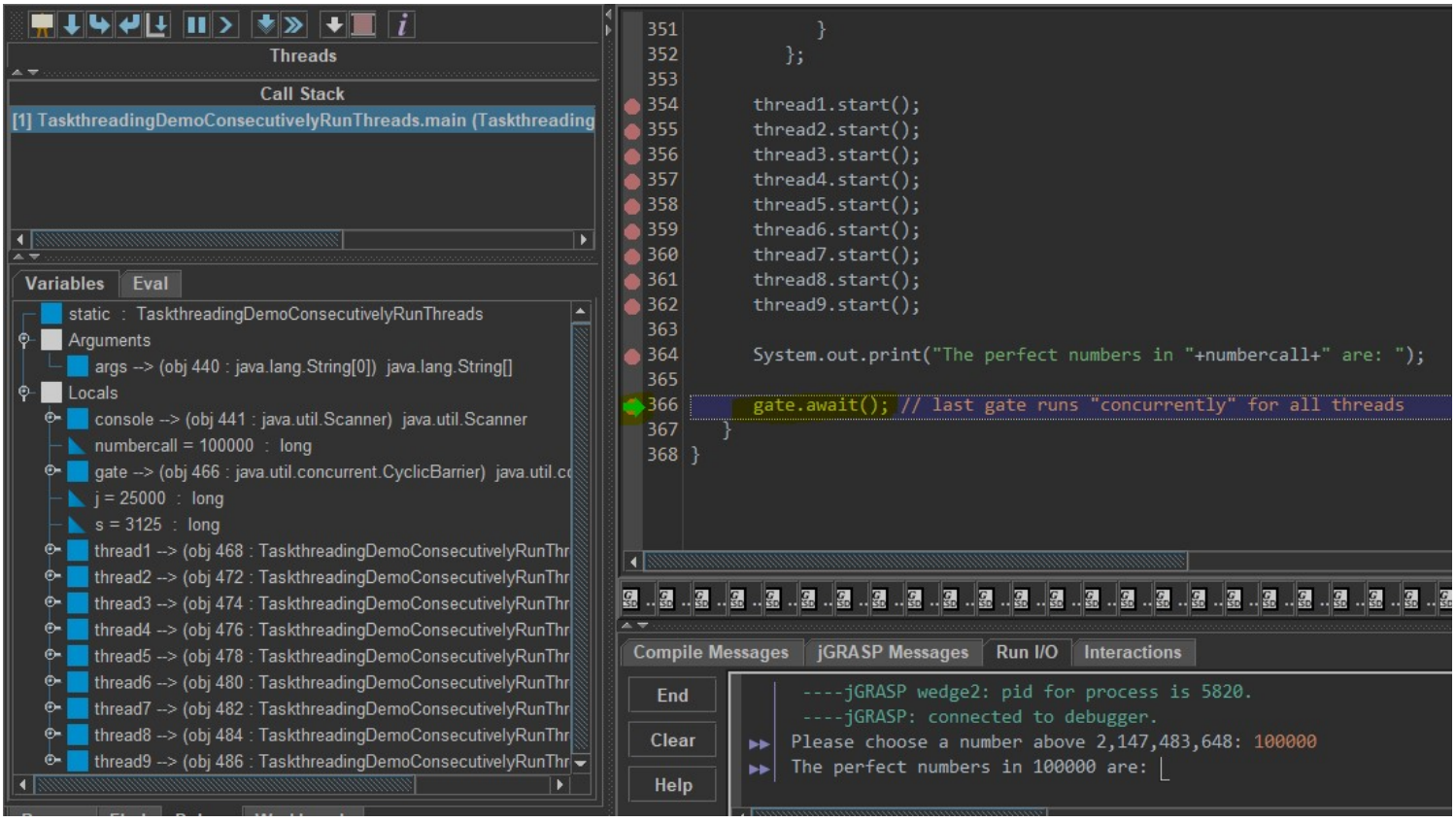
==========



==========

Remember we set our gate value to 10 so once we hit that last gate value, the gate threashold is reached and Java will release all the gates at once, it wil simply juggle all of this at once. WOW. I wish I could juggle like this. The threads will all start at the same time. Each thread#.start(); will start the thread however each has a gate.await() inside so none will really start until the last gate.await has been reached. Debug will reflect this as no number have printed out unit after gate.await 10.

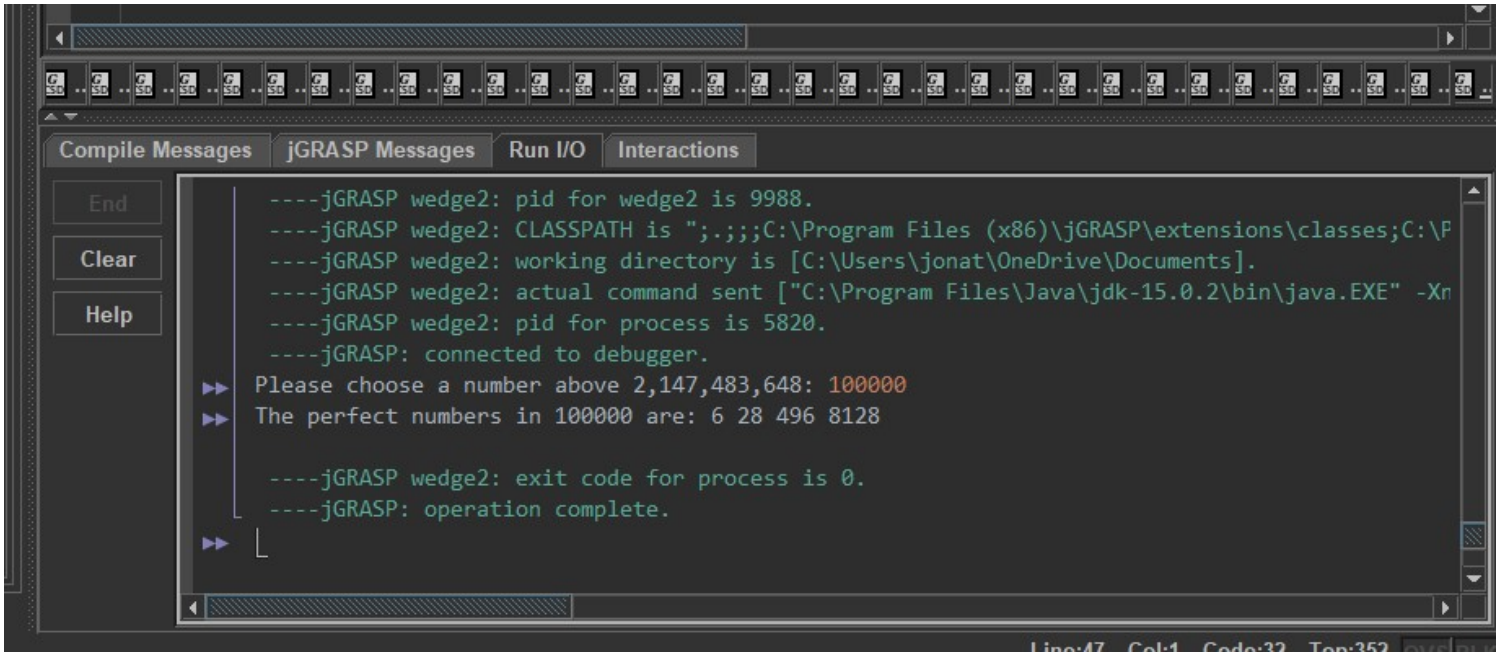
==========



==========

gate.await() 10 once I hit enter the threads will run concurrently and display the findings.

==========



==========

We made Java juggle 9 things at once. Amazing right? This is a basic understanding of multithreading. This took so much time and juggling to figure out I hope you enjoy this basic overview of Java's thread class and gates.

==========

Works Cited

94. M. (n.d.). Retrieved October 14, 2021, from https://www.cryptomuseum.com/crypto/usa/m94/index.htm.

A Christmas story. (1983). [DVD].

Liang, Y. D. (2011). *Introduction to java programming: Comprehensive version*. Prentice Hall.

Levy, S. (2009, April 20). *Mission impossible: The code even the CIA CAN'T CRACK*. Wired. Retrieved October 1, 2021, from https://www.wired.com/2009/04/ff-kryptos/.

Proc, J. (n.d.). M-94. Retrieved October 14, 2021, from http://www.jproc.ca/crypto/m94.html.

Schwartz, J., & Corum, J. (2020, January 29). *This sculpture holds a decades-old C.I.A. Mystery. and now, another clue.* The New York Times. Retrieved October 14, 2021, from https://www.nytimes.com/interactive/2020/01/29/climate/kryptos-sculpture-final-clue.html.

Zetter, K. (2014, November 21). *Finally, a new clue to solve the CIA's mysterious Kryptos Sculpture*. Wired. Retrieved October 14, 2021, from https://www.wired.com/2014/11/second-kryptos-clue/.