Breaking Code

Ada Lovelace Day

Code Poetry Event

October 12th, 12.00-13.00

IT UNIVERSITY OF COPENHAGEN

Ada Lovelace Day is an international day celebrating Ada Lovelace, the first computer programmer, and all achievements of women, transpeople and non-binaries in STEM

```
Event << We will have cake and make (and break) code poetry >>
Event << Celebratory vibes and creative attempts >>
// Neither booking nor experience is required.
```

ETHOS LAB

Ada Lovelace Day
Indie Event

Ada Lovelace and codes

Augusta Ada King (1815-1852), Countess of Lovelace, known as Ada Lovelace, published in 1843 as the first person what we would now call a computer programme. The programme was for mathematician’s Charles Babbage's Analytical Engine - the first mechanical computer - which only partly was built.

Ada wrote a set of instructions of how to calculate the numbers of Bernoulli on the Analytical Engine, encoding punch cards. She furthermore envisioned the Analytical Engine to be much more than just a smart calculator.

In mathematics, the Bernoulli numbers B_n are a sequence of rational numbers which occur frequently in number theory. The Bernoulli numbers were discovered independently by the Swiss mathematician **Jacob Bernoulli,** and the Japanese mathematician **Seki Takakazu.** Seki's discovery was posthumously published in **1712** in his work Katsuyō Sanpō; Bernoulli's, also posthumously, in his Ars Conjectandi of **1713.** Perhaps, not surprisingly, the discovery took the name after Jacob Bernoulli.

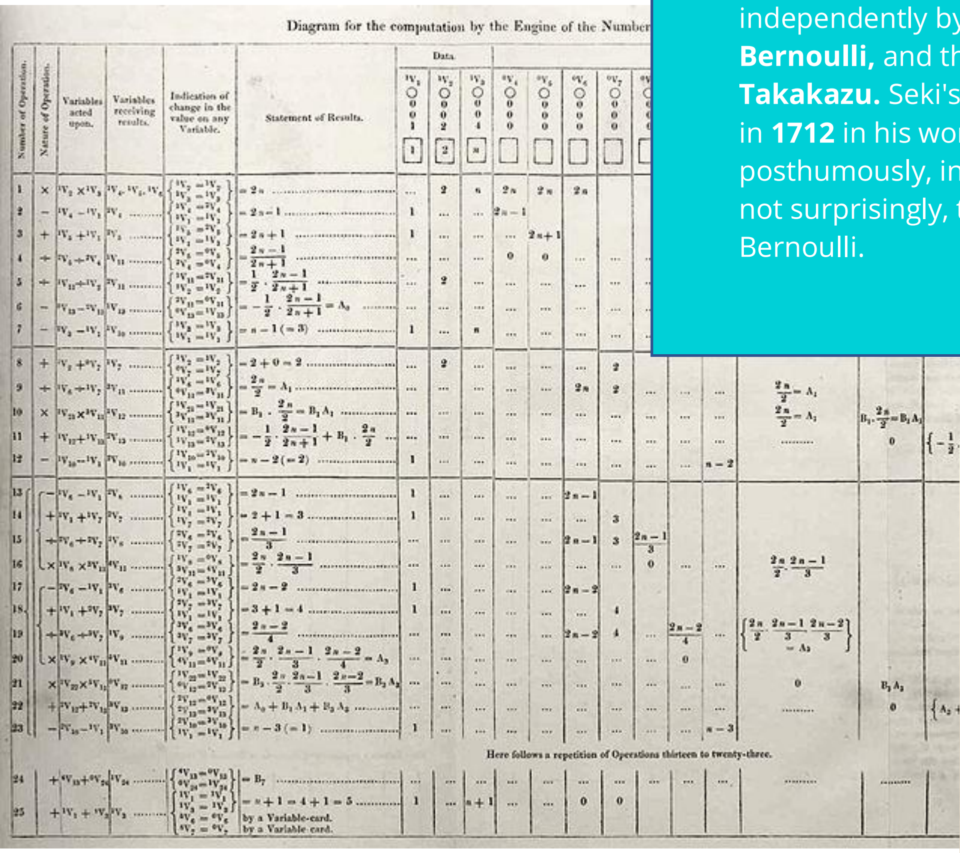


Diagram for the computation by the Engine of the Number

Here follows a repetition of Operations thirteen to twenty-three.

Ada related to the computer and to codes poetically, as well as mathematically and analytically. She saw that it could be making music or graphics, given the right inputs.

It would be another century before such computers were developed.

She foresaw what we now call computer science and further advocated for conducting science without having “usage” as the first thing in mind. Rather, she argued that insights will have effects and outcomes that never can be known from the point of ‘presence’. A comment which today feels ever so relevant.

> *Those who incline to very strictly utilitarian views may perhaps feel that the peculiar powers of the Analytical Engine bear upon questions of abstract and speculative science, rather than upon those involving every-day and ordinary human interests. These persons being likely to possess but little sympathy, or possibly acquaintance, with any branches of science which they do not find to be useful (according to their definition of that word), may conceive that the undertaking of that engine, now that the other one is already in progress, would be a barren and unproductive laying out of yet more money and labour; in fact, a work of supererogation*

Lovelace in: Menabrea, L. F., & Lovelace, A. (1842). Sketch of the analytical engine invented by Charles Babbage.

Ada Lovelace was a Victorian high society lady and those privileges enabled her to break from the common norms of what women *can* and *should be.* Less privileged women would not have been able to transgress the norms and fields as she, nor have the time or money to acquire the knowledge. Yet, she was experiencing a significant amount of resistance, facing gender discrimination. Her tutor Augustus De Morgan wrote a letter to her mother in which he questioned whether a women’s mind would be apt for doing math as that advanced level:

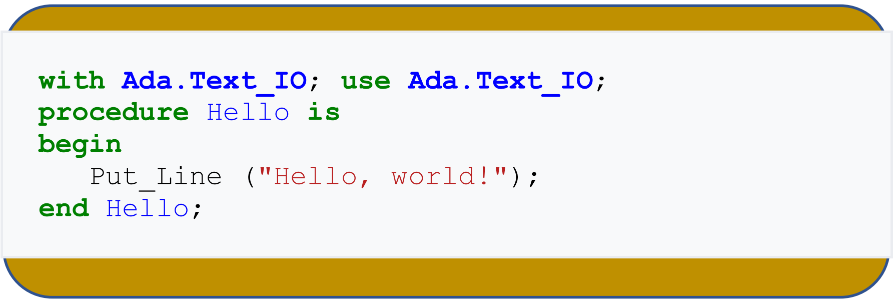
> *The very great tension of mind which they require is beyond the strength of a woman’s physical power of application.*

Dep. Lovelace-Byron (Bodleian Library, Oxford), Box 339, De Morgan to Lady Byron, 21 January 1844, ff 1-2

At the age of 36, Ada Lovelace passed away due to cancer. Sadly, the legacy she left behind wasn’t yet appreciated.

In the 1950s, Ada Lovelace's story picked back up when a man named B.V. Bowden rediscovered her contributions to the Analytical Engine's documentation.

From 1977 to 1983, a programming language was developed for the U.S. Department of Defense (DoD) carrying the name Ada. It was designed by a team led by French computer scientist Jean Ichbiah of CII Honeywell Bull intended to supersede over 450 programming languages used by the US DoD at that time.



```ada
with Ada.Text_IO; use Ada.Text_IO;
procedure Hello is
begin
   Put_Line ("Hello, world!");
end Hello;
```

Ada: Hello World

Matters of inequality within STEM does not merely concern a binary gender division, but matters of class, racialization, queer identities etc. Today, people's contributions are still overlooked, and some do not get the opportunities they deserve.

Ada Lovelace Day is a day of remembering just that as well as honoring and celebrating the scientific contributions by women, transpeople and non-binaries!

Code poetry – how?

As Ada quickly realized, codes and computers can create art. They can be artful.

> *(…) the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves.*
>
> Lovelace in: Menabrea, L. F., & Lovelace, A. (1842). Sketch of the analytical engine invented by Charles Babbage.

A code poem is a play of integrating languages. It can be interactive or static, digital or analog. A code poem does not need to comply to any syntax, but it can if you want it to.

Some code poems can run by computers, other can be performed by humans through spoken word and written text, and yet some are hybrid; being able of being read by both humans and computers.

Creating code poetry

Today, we will relate to codes not in terms of their functionality as a programming language but as a poetic language. We will make code poems like we want them to be, in relationship to our abilities, our voice, and our desires.

You may either do your poem analogue or digitally, and there are different materials available for you to mix and match:

- Feminist quotes
- A runnable code
- Code syntax symbols for clip out

It does not need to be genius (is that even a thing?) but a fun activity.
There can be joy in relating to code!

Analogue

Grab what you like to mix and match! It is encouraged to create, break, play with language (code and non-code). Sprinkle on some glitter, draw a diagram, cut out words or write your own.

React to what you would like to bring forward.

Digital

If you are not a coding star (yet), a good way to create a machine readable code poem is it utilize a resource from Khan Academy, where you write a poem in a poem ready terminal. When you are done with your poem, press the share button and you can have it send to your email.

- **https://www.khanacademy.org/computing/computer-programming/html-css/intro-to-html/pc/challenge-write-a-poem**

If you fare well in the terrain of codes, go ahead a be creative in a terminal of your choice. You can do beautiful shapes, twist common codes, or play with pattens.

Feel free to use, break, or connect with this code poem made by ETHOS Lab TA Edith Terte Andersen:

- https://drive.google.com/drive/folders/13U7KhAwtM9GytvqYI5pPQEG3ZKObb4oI?usp=sharing

Exhibit your code poetry

We would love to see your poems and share it with others!

Get your code poem on our Instagram

Send a picture of your code poem to our Instagram account: **ethositu**
Or mail: ethos@itu.dk

Get your code poem published

We want to create a code poetry booklet, based on code poems from this event.

If you would like to have your poem featured in the booklet, you should write us a mail with:

1) a high-resolution picture of your poem and the digital file (if there is one)
2) a statement if you would like your name to be featured or not
3) 2-4 lines reflection note about your poem to set the stage

You may continue workshopping your poem after this workshop or create a new one if you get into the loop. But note that the poem may be raw.

The deadline for contributions is November 1st 2021.
Write to: ethos@itu.dk

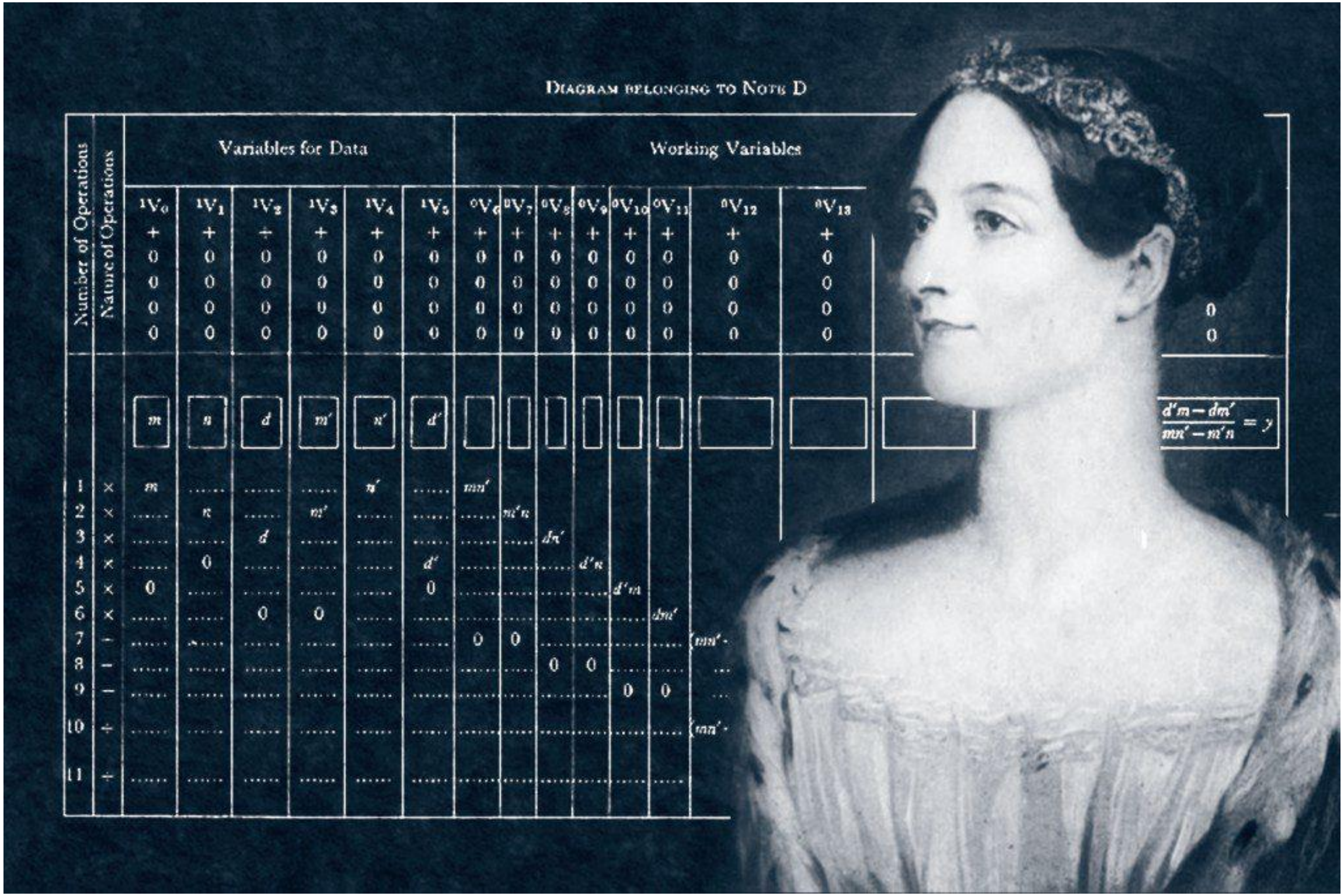


DIAGRAM BELONGING TO NOTE D

Number of Operations	Nature of Operations	Variables for Data						Working Variables									
		$^1V_0$	$^1V_1$	$^1V_2$	$^1V_3$	$^1V_4$	$^1V_5$	$^0V_6$	$^0V_7$	$^0V_8$	$^0V_9$	$^0V_{10}$	$^0V_{11}$	$^0V_{12}$	$^0V_{13}$		
		+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0		0 0
		$\boxed{m}$	$\boxed{n}$	$\boxed{d}$	$\boxed{m'}$	$\boxed{n'}$	$\boxed{d'}$										$\boxed{\frac{d'm-dm'}{mn'-m'n}=y}$
1	×	m	......	......	......	n'	......	mn'									
2	×	......	n	......	m'	......	......	......	$m'n$								
3	×	......	......	d	......	......	......	......	......	dn'							
4	×	......	0	......	......	......	d'	......	......	......	$d'n$						
5	×	0	......	......	......	......	0	......	......	......	......	$d'm$					
6	×	......	......	0	0	......	......	......	......	......	......	......	dm'				
7	−	......	......	......	......	......	......	0	0	......	......	......	......	$(mn'$-			
8	−	......	......	......	......	......	......	......	......	0	0	......	......	...			
9	−	......	......	......	......	......	......	......	......	......	......	0	0	...			
10	÷	......	......	......	......	......	......	......	......	......	......	......	......	$(mn'$-			
11	÷	......	......	......	......	......	......	......	......	......	......	......	......				