

# *Ada Lovelace and The Very First Computer Program*

ffconf

10<sup>th</sup> November 2023

by Steven Goodwin

@marquis de geek

[www.MarquisdeGeek.com](http://www.MarquisdeGeek.com)



@marquis de geek

---

---

# *Introduction*

- What was the first program?
- How does it work?
- Transform the original into JavaScript
- Was *Ada really* the first programmer?

*Who am I?*

@marquis de geek

---

---

# *The Ego Slide*

# Who am I?

- Game developer (C/C++/Assembler/JavaScript/AS)
  - PC, Xbox, Playstation, Gamecube, Wii
  - Online, mobile
- EdTech Entrepreneur and CTO
  - JavaScript
  - Serverless
- Emerging Technology Specialist
  - AR/VR/Alexa/Leap Motion
- Open source advocate
  - FOSDEM
  - Keynote speaker

# Who am I?

- Game developer (C/C++/Assembler/JavaScript/AS)
  - PC, Xbox, Playstation, Gamecube, Wii
  - Online, mobile
- EdTech Entrepreneur and CTO
  - JavaScript
  - Serverless
- Emerging Technology Specialist
  - AR/VR/Alexa/Leap Motion
- Open source advocate
  - FOSDEM
  - Keynote speaker



@marquis de geek

*What was the first program?*





# What does it do?

- Calculates Bernoulli numbers

$n$	0	1	2	4	6	8	10	12	14	16	18	20
$B_n$	1	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{30}$	$\frac{1}{42}$	$-\frac{1}{30}$	$\frac{5}{66}$	$-\frac{691}{2730}$	$\frac{7}{6}$	$-\frac{3617}{510}$	$\frac{43867}{798}$	$-\frac{174611}{330}$
$\frac{B_n}{n}$			$\frac{1}{12}$	$-\frac{1}{120}$	$\frac{1}{252}$	$-\frac{1}{240}$	$\frac{1}{132}$	$-\frac{691}{32760}$	$\frac{1}{12}$	$-\frac{3617}{8160}$	$\frac{43867}{14364}$	$-\frac{174611}{6600}$



# *How are the numbers calculated?*

- Stuff with fractions

# How are the numbers calculated?

- Stuff with fractions

$$0 = -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \binom{2n}{2} + B_3 \left( \frac{2n \cdot (2n-1) \cdot (2n-2)}{2 \cdot 3 \cdot 4} \right) \\ + B_5 \left( \frac{2n \cdot (2n-1) \cdot (2n-2) \cdot (2n-3) \cdot (2n-4)}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6} \right) + \dots + B_{2n-1}$$

# How are the numbers calculated?

- Stuff with fractions

$$0 = A_0(n) + A_1(n) \cdot B_1 + A_3(n) \cdot B_3 + \dots + A_{2n-3}(n) \cdot B_{2n-3} + B_{2n-1}$$

$$A_0(n) = -\frac{1}{2} \cdot \frac{2n-1}{2n+1}$$

$$A_1(n) = \frac{2n}{2}$$

$$A_3(n) = A_1(n) \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4}$$

$$A_5(n) = A_3(n) \cdot \frac{2n-3}{5} \cdot \frac{2n-4}{6}$$

# How are the numbers calculated?

- Stuff with fractions

$$-B_{2n-1} = A_0(n) + A_1(n) \cdot B_1 + A_3(n) \cdot B_3 + \dots + A_{2n-3}(n) \cdot B_{2n-3}$$

$$-B_5 = A_0(3) + A_1(3) \cdot B_1 + A_3(3) \cdot B_3$$

$$-B_7 = A_0(4) + A_1(4) \cdot B_1 + A_3(4) \cdot B_3 + A_5(4) \cdot B_5$$

*So how does the program do it?*





# So how does the program do it?

- Number of operation
- Nature of operation
- Variables acted upon
- Variables receiving results
- Indication of change in the value on any variable
- Statement of results

1	$\times$	$V_2 \times V_3$	$V_4, V_5, V_6$	$\left. \begin{array}{l} V_2 = V_2 \\ V_3 = V_3 \\ V_4 = V_4 \\ V_5 = V_5 \\ V_6 = V_6 \end{array} \right\}$	$= 2^N \dots\dots\dots$
2	$-$	$V_2 - V_3$	$V_4, \dots\dots\dots$	$\left. \begin{array}{l} V_2 = V_2 \\ V_3 = V_3 \\ V_4 = V_4 \\ V_5 = V_5 \\ V_6 = V_6 \end{array} \right\}$	$= 2^N - 1 \dots\dots\dots$
3	$+$	$V_2 + V_3$	$V_4, \dots\dots\dots$	$\left. \begin{array}{l} V_2 = V_2 \\ V_3 = V_3 \\ V_4 = V_4 \\ V_5 = V_5 \\ V_6 = V_6 \end{array} \right\}$	$= 2^N + 1 \dots\dots\dots$

# *Initialisation?*



# Initialisation?

- $V_1 = 1$
- $V_2 = 2$
- $V_3 = n = 4$

	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$
	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○
	1	2	n				
.....	...	2	n	2n	2n	2n	
.....	1	...	...	2n-1			
.....	1	...	...	...	2n+1		
.....	...	...	...	0	0	...	...
.....	...	2	...	...	...	...	...

# The very first computer program

1	x	$V_2 \times V_3$	$V_4, V_5, V_6$	$V_2 = V_2$	$= 2n$
2	-	$V_4 - V_1$	$2V_2$	$V_3 = 2V_3$	$= 2n - 1$
3	+	$V_2 + V_1$	$2V_2$	$V_4 = 2V_4$	$= 2n + 1$
4	+	$2V_2 + 2V_4$	$2V_{11}$	$2V_5 = 2V_5$	$= \frac{2n - 1}{2n + 1}$
5	+	$2V_{11} + V_2$	$2V_{11}$	$2V_6 = 2V_6$	$= \frac{2n - 1}{2n + 1}$
6	-	$2V_{12} - 2V_{11}$	$2V_{12}$	$2V_{11} = 2V_{11}$	$= \frac{1}{2} \cdot \frac{2n - 1}{2n + 1} = A_0$

# The very first computer program

1	X	${}^1V_2 \times {}^1V_3$	${}^1V_4, {}^1V_5, {}^1V_6$	$\left\{ \begin{array}{l} {}^1V_2 = {}^1V_2 \\ {}^1V_3 = {}^1V_3 \end{array} \right\}$	$= 2n \dots\dots\dots$
2	-	${}^1V_4 - {}^1V_1$	${}^2V_4$ .....	$\left\{ \begin{array}{l} {}^1V_4 = {}^2V_4 \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= 2n - 1 \dots\dots\dots$
3	+	${}^1V_5 + {}^1V_1$	${}^2V_5$ .....	$\left\{ \begin{array}{l} {}^1V_5 = {}^2V_5 \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= 2n + 1 \dots\dots\dots$
4	÷	${}^2V_5 \div {}^2V_4$	${}^1V_{11}$ .....	$\left\{ \begin{array}{l} {}^2V_5 = {}^0V_5 \\ {}^2V_4 = {}^0V_4 \end{array} \right\}$	$= \frac{2n - 1}{2n + 1} \dots\dots\dots$
5	÷	${}^1V_{11} \div {}^1V_2$	${}^2V_{11}$ .....	$\left\{ \begin{array}{l} {}^1V_{11} = {}^2V_{11} \\ {}^1V_2 = {}^1V_2 \end{array} \right\}$	$= \frac{1}{2} \cdot \frac{2n - 1}{2n + 1} \dots\dots\dots$
6	-	${}^0V_{13} - {}^2V_{11}$	${}^1V_{13}$ .....	$\left\{ \begin{array}{l} {}^2V_{11} = {}^0V_{11} \\ {}^0V_{13} = {}^1V_{13} \end{array} \right\}$	$= -\frac{1}{2} \cdot \frac{2n - 1}{2n + 1} = A_0$
7	-	${}^1V_3 - {}^1V_3$	${}^1V_3$	$\left\{ {}^1V_3 = {}^1V_3 \right\}$	$= n - 1 (= 3) \dots\dots\dots$

# The very first computer bug

1	X	${}^1V_2 \times {}^1V_3$	${}^1V_4, {}^1V_5, {}^1V_6$	$\left\{ \begin{array}{l} {}^1V_2 = {}^1V_2 \\ {}^1V_3 = {}^1V_3 \end{array} \right\}$	$= 2^n \dots\dots\dots$
2	-	${}^1V_4 - {}^1V_1$	${}^2V_4$ .....	$\left\{ \begin{array}{l} {}^1V_4 = {}^2V_4 \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= 2^n - 1 \dots\dots\dots$
3	+	${}^1V_5 + {}^1V_1$	${}^2V_5$ .....	$\left\{ \begin{array}{l} {}^1V_5 = {}^2V_5 \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= 2^n + 1 \dots\dots\dots$
4	÷	${}^2V_5 \div {}^2V_4$	${}^1V_{11}$ .....	$\left\{ \begin{array}{l} {}^2V_5 = {}^0V_5 \\ {}^2V_4 = {}^0V_4 \end{array} \right\}$	$= \frac{2^n - 1}{2^n + 1} \dots\dots\dots$
5	÷	${}^1V_{11} \div {}^1V_2$	${}^2V_{11}$ .....	$\left\{ \begin{array}{l} {}^1V_{11} = {}^2V_{11} \\ {}^1V_2 = {}^1V_2 \end{array} \right\}$	$= \frac{1}{2} \cdot \frac{2^n - 1}{2^n + 1} \dots\dots\dots$
6	-	${}^0V_{13} - {}^2V_{11}$	${}^1V_{13}$ .....	$\left\{ \begin{array}{l} {}^2V_{11} = {}^0V_{11} \\ {}^0V_{13} = {}^1V_{13} \end{array} \right\}$	$= -\frac{1}{2} \cdot \frac{2^n - 1}{2^n + 1} = A_0$
7	-	${}^1V_3 - {}^1V_3$	${}^1V_3$	$\left\{ {}^1V_3 = {}^1V_3 \right\}$	$= 0 (= 3) \dots\dots\dots$

# A basic simulator

```
/* 1 */ { op: 'x', a:2, b:3, d:[4,5,6] }, // 2n
/* 2 */ { op: '-', a:4, b:1, d:[4] }, // 2n-1
/* 3 */ { op: '+', a:5, b:1, d:[5] }, // 2n+1
/* 4 */ { op: '/', a:4, b:5, d:[11] }, // (2n-1) / (2n+1)
/* 5 */ { op: '/', a:11, b:2, d:[11] }, // (2n-1) / (2n+1) / 2
```



# A basic simulator

```
let a = code[pc].a;
let b = code[pc].b;

switch (code[pc].op) {
  case 'x':
    mill = state.v[a] * state.v[b];
    break;
  case '+':
    mill = state.v[a] + state.v[b];
    break;
  case '-':
    mill = state.v[a] - state.v[b];
    break;
  case '/':
    mill = state.v[a] / state.v[b];
    break;
}
```

# *A basic simulator*

```
code[pc].d.forEach(function(dest) {  
    state.v[dest] = null;  
});
```

```
pc = pc + 1;
```

# Going around?

- Remember this?

$$-B_5 = A_0(3) + A_1(3) \cdot B_1 + A_3(3) \cdot B_3$$

$$-B_7 = A_0(4) + A_1(4) \cdot B_1 + A_3(4) \cdot B_3 + A_5(4) \cdot B_5$$

# Going around?

- The first loop!

21	$\times$	${}^1V_{22} \times {}^5V_{11}$	${}^0V_{12}$ .....	$\left\{ \begin{array}{l} {}^1V_{22} = {}^1V_{22} \\ {}^0V_{12} = {}^2V_{12} \end{array} \right\}$	$= B_3 \cdot \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = B_3 A_3$	...	...	...	...	...	...	...	...	...	...	...	...	0
22	$+$	${}^2V_{12} + {}^2V_{13}$	${}^3V_{13}$ .....	$\left\{ \begin{array}{l} {}^2V_{12} = {}^0V_{12} \\ {}^2V_{13} = {}^3V_{13} \end{array} \right\}$	$= A_0 + B_1 A_1 + B_3 A_3$ .....	...	...	...	...	...	...	...	...	...	...	...	...	.....
23	$-$	${}^2V_{10} - {}^1V_1$	${}^3V_{10}$ .....	$\left\{ \begin{array}{l} {}^2V_{10} = {}^3V_{10} \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= n - 3 (= 1)$ .....	1	...	...	...	...	...	...	...	...	...	...	...	$n - 3$

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

24	$+$	${}^4V_{13} + {}^0V_{24}$	${}^1V_{24}$ .....	$\left\{ \begin{array}{l} {}^4V_{13} = {}^0V_{13} \\ {}^0V_{24} = {}^1V_{24} \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= B_7$ .....	...	...	...	...	...	...	...	...	...	...	...	...	.....
----	-----	---------------------------	--------------------	---	---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-------

*But...*

*But...*

- The variables are constant!

# The variables are constant!

- Always a given number!
- No way of saying  $v[idx]$

21	$\times$	${}^1V_{22} \times {}^5V_{11}$	${}^0V_{12}$ .....	$\left\{ \begin{array}{l} {}^1V_{22} = {}^1V_{22} \\ {}^0V_{12} = {}^2V_{12} \end{array} \right\}$	$= B_3 \cdot \frac{2^n}{2} \cdot \frac{2^{n-1}}{3} \cdot \frac{2^{n-2}}{3} = B_3 A_3$	...	...	...	...	...	...	...	...	...	...	...	...	0
22	$+$	${}^2V_{12} + {}^2V_{13}$	${}^3V_{13}$ .....	$\left\{ \begin{array}{l} {}^2V_{12} = {}^0V_{12} \\ {}^2V_{13} = {}^3V_{13} \end{array} \right\}$	$= A_0 + B_1 A_1 + B_3 A_3$ .....	...	...	...	...	...	...	...	...	...	...	...	...	.....
23	$-$	${}^2V_{10} - {}^1V_1$	${}^3V_{10}$ .....	$\left\{ \begin{array}{l} {}^2V_{10} = {}^3V_{10} \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= n - 3 (= 1)$ .....	1	...	...	...	...	...	...	...	...	...	...	...	$n - 3$

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

24	$+$	${}^4V_{13} + {}^0V_{24}$	${}^1V_{24}$ .....	$\left\{ \begin{array}{l} {}^4V_{13} = {}^0V_{13} \\ {}^0V_{24} = {}^1V_{24} \\ {}^1V_1 = {}^1V_1 \end{array} \right\}$	$= B_7$ .....	...	...	...	...	...	...	...	...	...	...	...	...	.....
----	-----	---------------------------	--------------------	---	---------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-------

# The transformation process

- `{ op:'x', a:2, b:3, d:[4,5,6] }, // 2n`
- `state.v[4] = state.v[5] = state.v[6] = state.v[2] * state.v[3];`
- `state.v[N2M1] = state.v[N2P1] = state.v[6] = 2 * state.v[N];`
  
- `{ op:'-', a:4, b:1, d:[4] }, // 2n-1`
- `state.v[4] -= 1;`
- `state.v[N2M1] -= 1;`
  
- `accumulatingTotal = (2*n - 1)`

<https://github.com/MarquisdeGeek/Ada-Origins>

@marquis de geek



# The transformation process

```
let indexToResult = 1;
do {
  accumulatingFraction *= (numerator - 1) / (denom + 1); // first terms
of A : (2n-1)/3
  accumulatingFraction *= (numerator - 2) / (denom + 2); // #19 second
2n+1/4

  accumulatingTotal += accumulatingFraction *
state.results[indexToResult]; // #21 A3 * B3

  numerator -= 2;
  denom += 2;

  ++indexToResult;
} while(--k > 0);

state.results[indexToResult] = -accumulatingTotal;
```

@marquis de geek

# *Was Ada really the first programmer?*

- The code has a bug
- The notation is problematic
- The loop didn't work
- She was translating someone else's work
- Babbage had already written programs for his own machine

# *Was Ada really the first programmer?*

- I have only one more argument...

# *Was Ada really the first programmer?*

- I have only one more argument...
- ...why *should* she be the first?

# *Was Ada really the first programmer?*

- If this work had been done by a man, would he be considered the first ever programmer?

# *Was Ada really the first programmer?*

- If this work had been done by a man, would he be considered the first ever programmer?
- YES!

## *Was Ada really the first programmer?*

- If this work had been done by a man, would he be considered the first ever programmer?
- YES!
- Therefore, Ada was the first programmer!





# Conclusions

- The first program did decimal approximations of Bernoulli numbers
- The algorithm given described what to do, but not how
- Ada was that first programmer

*Any Questions? Ask me later!*

@marquis de geek

www.MarquisdeGeek.com

<https://github.com/MarquisdeGeek/Ada-Origins>

[https://marquisdegeek.com/code\\_ada99](https://marquisdegeek.com/code_ada99)



@marquis de geek

---

---