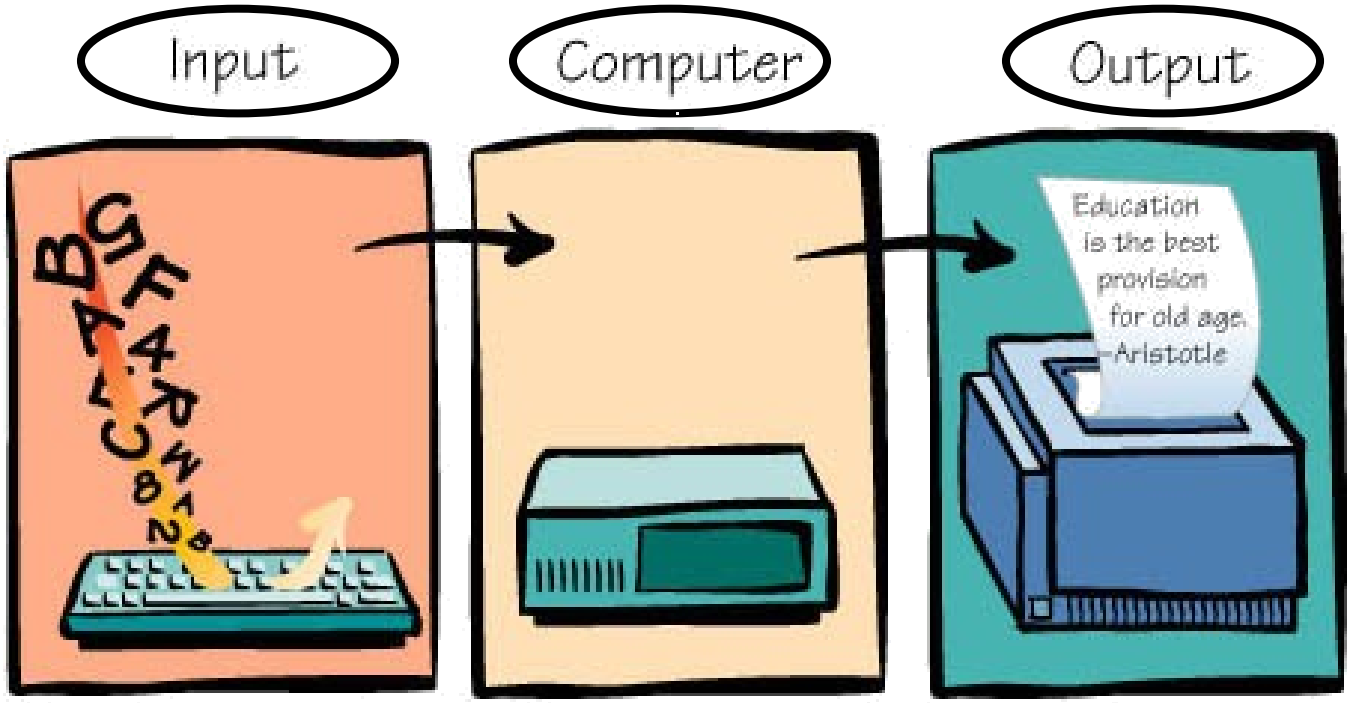
Od obliczeń do komunikacji

Jak do tego wszystkiego doszło?

Życie bez komputerów?

Komputery ingerują w nasze życie prawie wszędzie

Typowy przepływ informacji



Wprowadzanie dokumentów

Przetwa-rzanie

Drukowanie

Pierwsi informatycy

- W sposób uświadomiony metody algorytmiczne stosowali starożytni Grecy
- Opracowali całą teorię konstrukcji geometrycznych o całkiem praktycznych zastosowaniach

Pierwsi twórcy algorytmów

- Euklides (~365-~300pne): algorytm obliczania największego wspólnego dzielnika
- Eratostenes (~275-~194pne): algorytm wyznaczania liczb pierwszych – sito Eratostenesa

Wyznaczanie liczb pierwszych

- Liczba pierwsza, to taka liczba, która ma dokładnie dwa różne podzielniki: jedność i samą siebie, np 2,3,5,7,11,...`
- Wcale nie jest łatwo:
  - stwierdzić, czy liczba jest pierwsza
  - wyznaczyć kolejną liczbę pierwszą
  - wiedząc, że liczba nie jest pierwsza znaleźć jej dzielniki

Sito Eratostenesa

- Wypisz odpowiednio dużo liczb naturalnych poczynając od 2
- Wykreśl wszystkie liczby podzielne przez 2
- Kolejno pobieraj pierwsze niewykreślone liczby i wykreślaj ich wielokrotności.

Przykład – wypisujemy odpowiednio dużo kolejnych liczb

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53

  ...

Przykład – wykreślamy wielokrotności 2

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  18 19 20 21 22 23 24 25 26 27 28 29
  30 31 32 33 34 35 36 37 38 39 40 41
  42 43 44 45 46 47 48 49 50 51 52 53

  ...

Przykład – wykreślamy wielokrotności 3

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  18 19 20 21 22 23 24 25 26 27 28 29
  30 31 32 33 34 35 36 37 38 39 40 41
  42 43 44 45 46 47 48 49 50 51 52 53

  ...

Przykład – wykreślamy wielokrotności 5

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  18 19 20 21 22 23 24 25 26 27 28 29
  30 31 32 33 34 35 36 37 38 39 40 41
  42 43 44 45 46 47 48 49 50 51 52 53

  ...

Przykład – wykreślamy wielokrotności 7

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  18 19 20 21 22 23 24 25 26 27 28 29
  30 31 32 33 34 35 36 37 38 39 40 41
  42 43 44 45 46 47 48 49 50 51 52 53

  ...

Zostały już tylko liczby pierwsze

Problemy z algorytmem Eratostenesa

- Nadaje się tylko do znajdowania stosunkowo niewielkich liczb
- Wymaga dodatkowej pamięci na przechowywanie kolejnych liczb naturalnych – jest złożony pamięciowo.

Największa liczba pierwsza do dziś odkryta

- $2^{82,589,933}-1$ jest nawiększą do 13.10.2008 znaną liczbą pierwszą. Została on odkryta w 07.12.2018 r. przez Patricka Lahore za pomocą GIMPS. Składa się ona z ok. 24,862,048 miliona cyfr.
- Użyta metoda jest zupełnie inna, niż za pomocą sita Eratostenesa.

Problem wyznaczenia największego wspólnego dzielnika

- Dane są dwie liczby naturalne m,n
- Znajdź największą liczbę naturalną, przez którą dzieli się bez reszty zarówno m jak i n.
- Na przykład dla m=18 i n=24 NWD(m,n)=6

Algorytm Euklidesa (1)

- 0<=m<=n, n>0
- Jeśli m=0 to NWD(m,n)=n
- Jeśli m>0 to NWD(m,n)=NWD(m,n-m)

Wada algorytmu Euklidesa (1)

- Jest bardzo wolny – przy złośliwych danych, czyli wtedy, gdy jedna z liczb będzie bardzo duża, a druga będzie jedynką, trzeba odejmować od większej z nich jedynkę tyle razy, ile wynosi wartość większej liczby.
- ... co już dla liczb 30-cyfrowych jest zupełnie niewykonalne!

Algorytm Euklidesa (2)

- 0<=m<=n, n>0
- Jeśli m=0 to NWD(m,n)=n
- Jeśli m>0 to

  NWD(m,n)=NWD(n mod m, m)

Gdzie n mod m to reszta z dzielenia n przez m

Wady i zalety

- Wada: programuje się trudniej (dzielenie trudniej zaprogramować niż odejmowanie)
- Zaleta: wykonuje się szybko – proporcjonalnie do długości zapisu liczby w systemie dziesiętnym.

Algorytm Euklidesa (3)

- Jeśli m=0 to NWD(n,m) =n
- Jeśli n=0 to NWD(n,m)=m
- Jeśli n,m \in P to NWD(n,m)=2NWD(n/2,m/2)
- Jeśli n jest parzysta, a m nieparzysta, to NWD(n,m)=NWD(n/2,m)
- Jeśli n jest nieparzysta, a m parzysta, to NWD(n,m)=NWD(n,m/2)
- Jeśli n,m są nieparzyste, to NWD(n,m)=NWD(n-m,m) dla n>=m lub NWD(n,m)=NWD(m-n,n) dla m>=n

Wady i zalety

- Wady:
  - ma skomplikowany opis
  - wykonuje się nieco dłużej niż poprzedni algorytm
- Zaleta: nadal wykonuje się szybko; proporcjonalnie do długości zapisu liczb.

Podsumowanie algorytmów Euklidesa

- Algorytm (1) jest bezużyteczny w praktyce, chyba że ktoś działa z małymi liczbami. Jego złożoność czasowa jest tak duża, że dla danych spotykanych np. w kryptografii nie bylibyśmy w stanie wykonać podstawowych kroków algorytmów szyfrujących.

Podsumowanie algorytmów Euklidesa -cd

- Algorytmy (2) i (3) są porównywalne, jeśli chodzi o jakość; częściej stosuje się algorytm (2).
- Pozwalają one swobodnie obliczać NWD nawet dla liczb kilkusetcyfrowych w rozsądnym czasie.

Dziedziny algorytmiczne

- Euklides 1: (N, =0 , <= , -)
- Euklides 2: (N, =0, mod)
- Euklides 3: (N, =0, <= , Par? ,*2, /2, -)

Przykład

- NWD(36,84) =NWD(36,48)=
  =NWD(36,12)=NWD(24,12)=
  =NWD(12,12)=NWD(12,0)=12
- NWD(36,84)=NWD(12,36)=NWD(0,12)=12
- NWD(36,84)=2*NWD(18,42)=4*NWD(9,21)=
  4*NWD(12,21)=4*NWD(6,21)=4*NWD(3,21)=
  4*NWD(3,18)=4*NWD(3,9)=
  4*NWD(3,6)=4*NWD(3,3)=4*NWD(0,3)=4*3
  =12

Al Chwarizmi

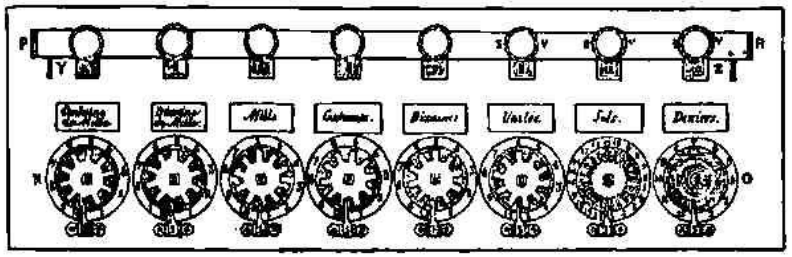
- Abu Ja'far Muhammad ibn Musa Al-Chuwarizmi (~780-~850)
- *Hisab al-jabr w'al-muqabala*
- *Opisał ciekawe algorytmy*

Wilhelm Schickard 1592-1635

- Zbudował pierwszy kalkulator

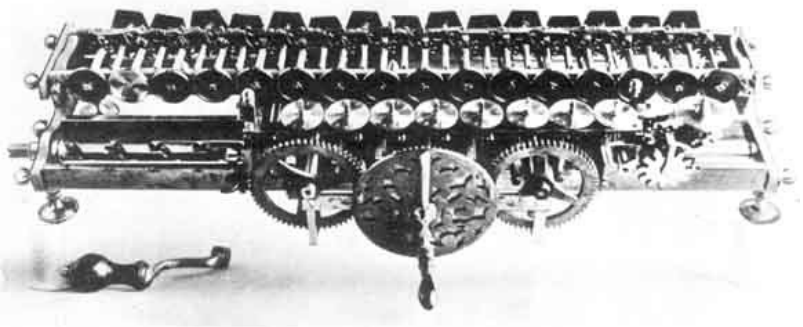
Blaise Pascal 1623-1662

- Wybitny matematyk, fizyk, teolog, filozof, konstruktor
- Między innymi zajmował się konstrukcją kalkulatorów



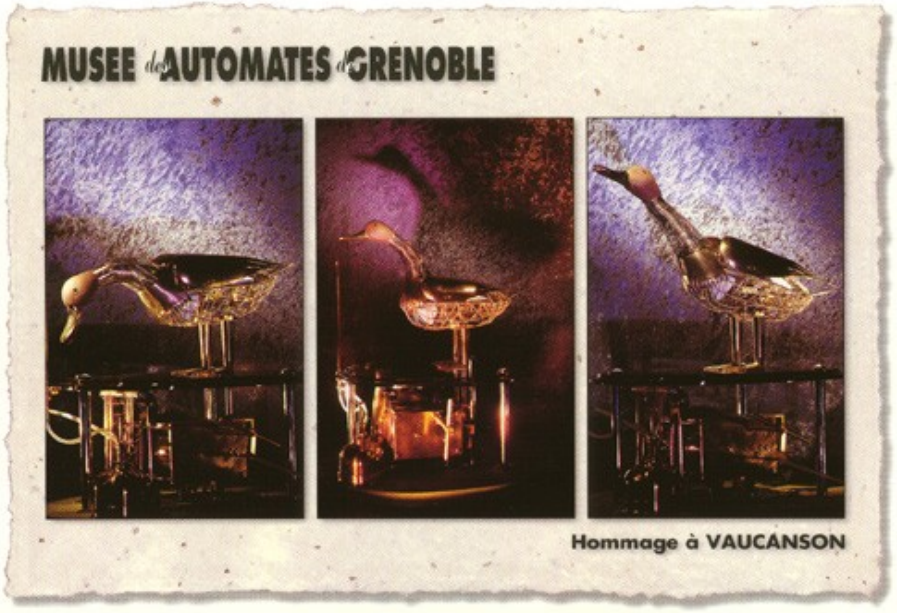
Gotfried Wilhelm Leibniz 1646-1716

- Wybitny filozof, matematyk, prawnik i konstruktor
- Udoskonalił kalkulator
- Wymyślił mechanizm zwany kołem Leibniza



Jacques de Vaucanson (1709-82)

- Wielki francuski konstruktor I wynalazca. Ojciec robotyki
- Skonstruował wiele mechanicznych zabawek
- Jako pierwszy w historii użył kart perforowanych do zapisu danych



Joseph Jacquard (1752-1834)

- twórca krosna tkackiego,
- wykorzystał pomysły de Vaucansona
- używał kart perforowanych do kodowania wzorów tkackich

Karty perforowane Jacquarda

Z tego zestawu

dziurek powstaje

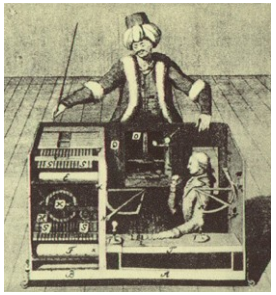
piękna tkanina

Farkas von Kempelen (1734-1804)

Wynalazca węgierski.
Twórca automatu szachowego
MEPHISTO

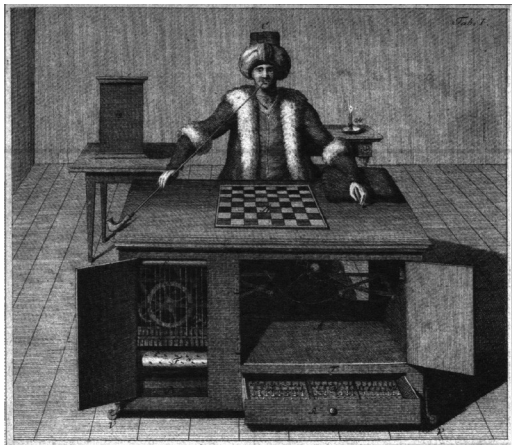
Farkas von Kempelen 1734-1804

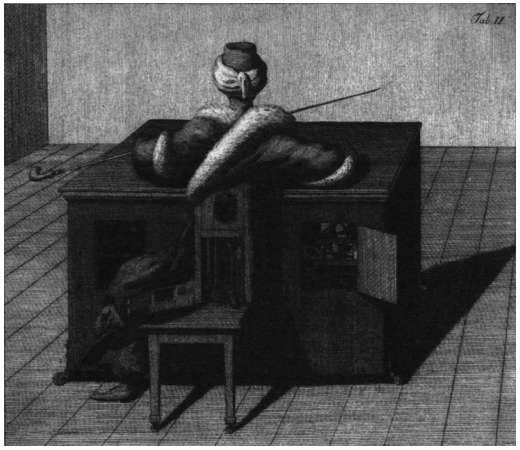
- Zbudował automat szachowy. Ruchy wykonywał ukryty szachista.



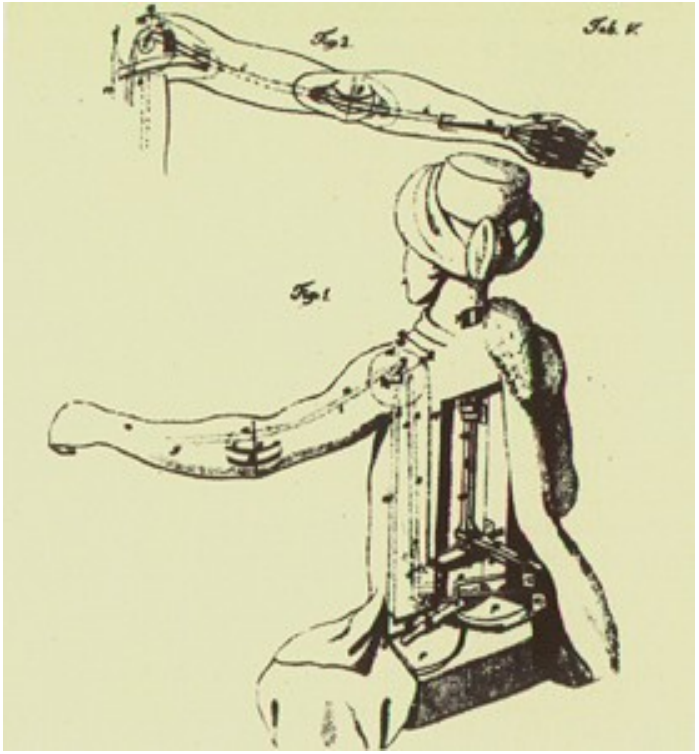
- Arcydzieło sztuki

Mephisto – Turek szachista von Kempelena





Mephisto (2)



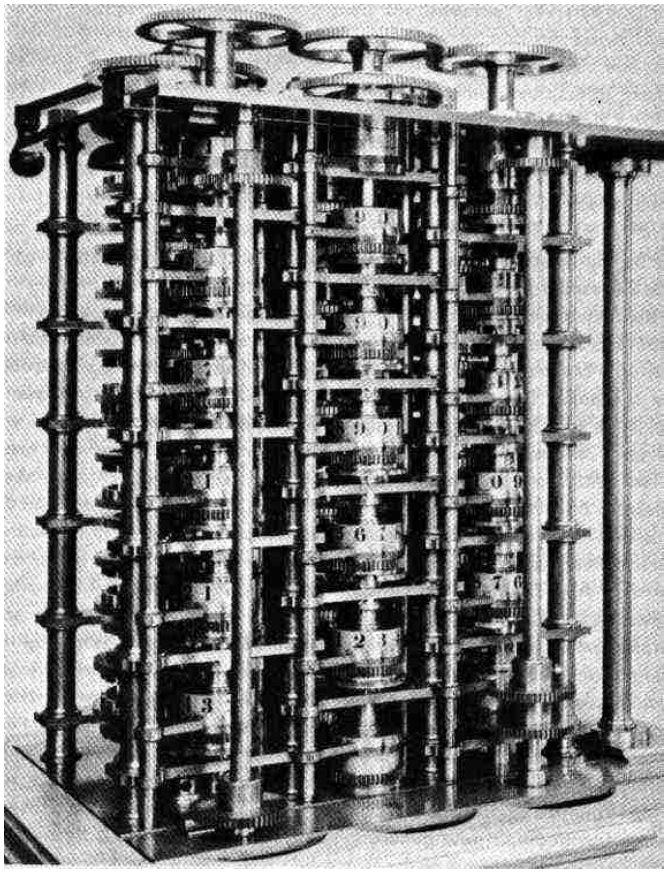
Abraham Stern (1749-1842)

- Jako pierwszy skonstruował kalkulator wyciągający pierwiastki

Charles Babbage (1791-1871)

- Twórca pierwszej maszyny liczącej
- Maszyna różnicowa
- Maszyna analityczna

Maszyna różnicowa



Joseph Jacquard

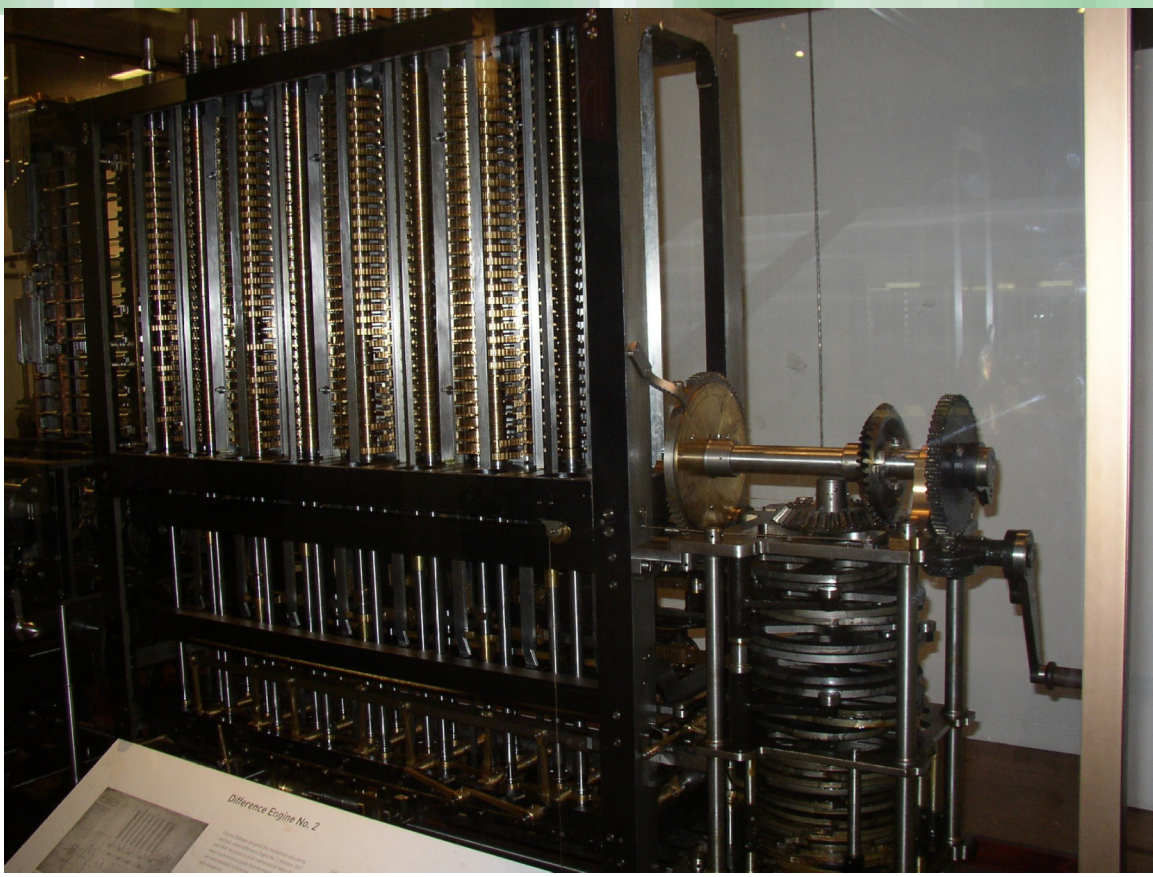
- Portret wynalazcy demonstrowany przez Babbage'a na organizowanych przez niego przyjęciach.

Maszyna różnicowa – wersja Babbage'a

- Rekonstrukcja pierwszej maszyny różnicowej.
- Ok. 1m wysokości

Maszyna różnicowa nr 2 – rekonstrukcja z 1990 r.

- Waży 15 ton
- Mierzy ok 4m wysokości
- Potrafi drukować wyniki



Ada Augusta Lovelace
1815-1852

Pierwsza programistka

w historii

Nowatorskie idee Ady Lovelace (1)

- Pomysł, że urządzenie może być programowalne i niekoniecznie wyspecjalizowane. Programy można wprowadzać na kartach perforowanych.

Nowatorskie idee Ady Lovelace (2)

- Pomysł, że przetwarzane nie muszą być tylko liczby. Liczby mogą kodować inne obiekty takie jak nuty, teksty, obrazy.

Nowatorskie idee Ady Lovelace (3)

- Przykład konkretnego programu liczącego liczby Bernoulliego

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data. ${}^1V_1$ 0 0 0 1 [1]	${}^1V_2$ 0 0 0 2 [2]	${}^1V_3$ 0 0 0 4 [n]	Working Variables. ${}^0V_4$	${}^0V_5$	${}^0V_6$	${}^0V_7$	${}^0V_8$	${}^0V_9$	${}^0V_{10}$	${}^0V_{11}$	${}^0V_{12}$	${}^0V_{13}$	Result Variables. ${}^1V_{21}$ B_1 is a decimal fraction.	${}^1V_{22}$ B_2 is a decimal fraction.	${}^1V_{23}$ B_3 is a decimal fraction.	${}^0V_{24}$... B_7
1	×	${}^1V_2 \times {}^1V_3$	${}^1V_4, {}^1V_5, {}^1V_6$	${}^1V_2 = {}^1V_2$ ${}^1V_3 = {}^1V_3$	$= 2n$	...	2	n	$2n$	$2n$	$2n$											
2	−	${}^1V_4 - {}^1V_1$	${}^2V_4$	${}^1V_4 = {}^2V_4$ ${}^1V_1 = {}^1V_1$	$= 2n-1$	1	...	...	$2n-1$													
3	+	${}^1V_5 + {}^1V_1$	${}^2V_5$	${}^1V_5 = {}^2V_5$ ${}^1V_1 = {}^1V_1$	$= 2n+1$	1	...	...	...	$2n+1$												
4	+	${}^2V_5 + {}^2V_4$	${}^1V_{11}$	${}^2V_5 = {}^0V_5$ ${}^2V_4 = {}^0V_4$	$= \frac{2n-1}{2n+1}$	...	...	...	0	0	...	...	...	...	...	$\frac{2n-1}{2n+1}$						
5	+	${}^1V_{11} + {}^1V_2$	${}^2V_{11}$	${}^1V_{11} = {}^2V_{11}$ ${}^1V_2 = {}^1V_2$	$= \frac{1}{2} \cdot \frac{2n-1}{2n+1}$	...	2	...	...	...	...	...	...	...	...	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$						
6	−	${}^0V_{13} - {}^2V_{12}$	${}^1V_{12}$	${}^2V_{11} = {}^0V_{11}$ ${}^0V_{13} = {}^1V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	...	...	...	...	...	...	...	...	...	...	0	.....	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$				
7	−	${}^1V_3 - {}^1V_1$	${}^2V_{10}$	${}^1V_3 = {}^1V_3$ ${}^1V_1 = {}^1V_1$	$= n-1\ (=3)$	1	...	n	...	...	...	...	...	...	$n-1$							
8	+	${}^2V_7 + {}^0V_7$	${}^1V_7$	${}^1V_7 = {}^1V_7$ ${}^0V_7 = {}^1V_7$	$= 2 + 0 = 2$	...	2	...	...	...	...	2										
9	÷	${}^1V_6 \div {}^1V_7$	${}^2V_{11}$	${}^1V_6 = {}^1V_6$ ${}^0V_{11} = {}^2V_{11}$	$= \frac{2n}{2} = A_1$	...	...	...	...	...	$2n$	2	...	...	...	$\frac{2n}{2} = A_1$						
10	×	${}^1V_{21} \times {}^2V_{11}$	${}^1V_{12}$	${}^1V_{21} = {}^1V_{21}$ ${}^2V_{11} = {}^2V_{11}$	$= B_1 \cdot \frac{2n}{2} = B_1 A_1$	...	...	...	...	...	...	...	...	...	...	$\frac{2n}{2} = A_1$	$B_1 \cdot \frac{2n}{2} = B_1 A_1$	.....	B_1			
11	+	${}^1V_{12} + {}^1V_{13}$	${}^2V_{13}$	${}^1V_{12} = {}^0V_{12}$ ${}^1V_{13} = {}^2V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$ ...	...	...	...	...	...	...	...	...	...	...	.....	0	$\left\{-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}\right\}$				
12	−	${}^1V_{10} - {}^1V_1$	${}^2V_{10}$	${}^1V_{10} = {}^2V_{10}$ ${}^1V_1 = {}^1V_1$	$= n-2\ (=2)$	1	...	...	...	...	...	...	...	...	$n-2$							
13	−	${}^1V_6 - {}^1V_1$	${}^2V_6$	${}^1V_6 = {}^2V_6$ ${}^1V_1 = {}^1V_1$	$= 2n-1$	1	...	...	...	...	$2n-1$											
14	+	${}^1V_1 + {}^1V_7$	${}^2V_7$	${}^1V_1 = {}^1V_1$ ${}^1V_7 = {}^2V_7$	$= 2+1 = 3$	1	...	...	...	...	...	3										
15	+	${}^2V_6 + {}^2V_7$	${}^1V_8$	${}^2V_6 = {}^2V_6$ ${}^2V_7 = {}^2V_7$	$= \frac{2n-1}{3}$	...	...	...	...	...	$2n-1$	3	$\frac{2n-1}{3}$									
16	×	${}^1V_8 \times {}^3V_{11}$	${}^4V_{11}$	${}^1V_8 = {}^0V_8$ ${}^3V_{11} = {}^4V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3}$	...	...	...	...	...	...	...	0	...	...	$\frac{2n}{2} \cdot \frac{2n-1}{3}$						
17	−	${}^2V_6 - {}^1V_1$	${}^3V_6$	${}^2V_6 = {}^3V_6$ ${}^1V_1 = {}^1V_1$	$= 2n-2$	1	...	...	...	...	$2n-2$											
18	+	${}^1V_1 + {}^2V_7$	${}^3V_7$	${}^2V_7 = {}^3V_7$ ${}^1V_1 = {}^1V_1$	$= 3+1 = 4$	1	...	...	...	...	...	4										
19	+	${}^3V_6 + {}^3V_7$	${}^4V_9$	${}^3V_6 = {}^3V_6$ ${}^3V_7 = {}^3V_7$	$= \frac{2n-2}{4}$	...	...	...	...	...	$2n-2$	4	...	$\frac{2n-2}{4}$	...	$\left\{\frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3}\right\}$ $= A_2$							
20	×	${}^1V_9 \times {}^4V_{11}$	${}^5V_{11}$	${}^1V_9 = {}^0V_9$ ${}^4V_{11} = {}^5V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = A_2$ ...	...	...	...	...	...	...	...	0									
21	×	${}^1V_{22} \times {}^5V_{11}$	${}^0V_{12}$	${}^1V_{22} = {}^1V_{22}$ ${}^0V_{12} = {}^2V_{12}$	$= B_3 \cdot \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = B_3 A_2$	...	...	...	...	...	...	...	...	...	...	0	$B_3 A_2$	.....	.....	B_2		
22	+	${}^2V_{12} + {}^2V_{12}$	${}^3V_{12}$	${}^2V_{12} = {}^0V_{12}$ ${}^2V_{12} = {}^3V_{12}$	$= A_0 + B_1 A_1 + B_2 A_2$	...	...	...	...	...	...	...	...	...	...	.....	0	$\left\{A_3 + B_1 A_1 + B_2 A_2\right\}$				
23	−	${}^2V_{10} - {}^1V_1$	${}^3V_{10}$	${}^2V_{10} = {}^3V_{10}$ ${}^1V_1 = {}^1V_1$	$= n-3\ (=1)$	1	...	...	...	...	...	...	...	...	$n-3$							

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

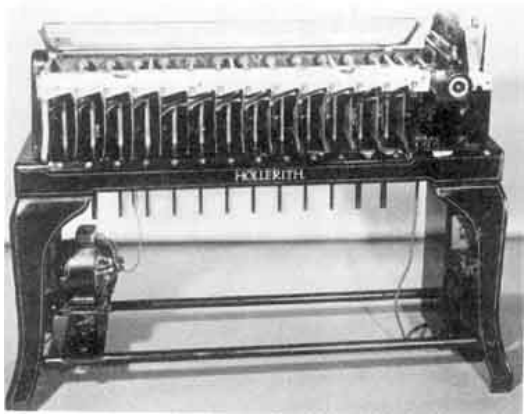
24	+	${}^4V_{12} + {}^0V_{24}$	${}^1V_{24}$	${}^4V_{12} = {}^0V_{12}$ ${}^0V_{24} = {}^1V_{24}$	$= B_7$	...	...	...	...	...	...	...	...	...	...	.....	.....	.....	.....	.....	.....	B_7
25	+	${}^1V_1 + {}^1V_2$	${}^1V_2$	${}^1V_1 = {}^1V_1$ ${}^1V_2 = {}^1V_2$ ${}^5V_6 = {}^0V_6$ ${}^5V_7 = {}^0V_7$	$= n+1 = 4+1 = 5$ ..... by a Variable-card. by a Variable card.	1	...	$n+1$	...	...	0	0										

Nowatorskie idee Ady Lovelace (4)

- Pomysł, że komputer może wykonywać czynność myślenia i samodzielnie podejmować decyzje.

Herman Hollerith 1860-1929

- Karty perforowane w przetwarzaniu danych
- Założyciel IBM



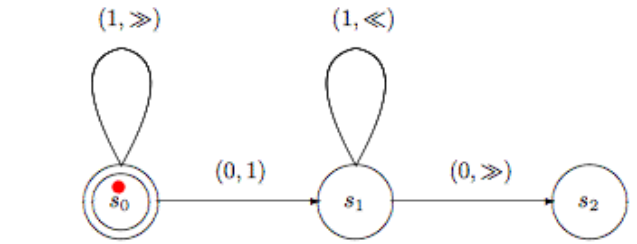
Alan Turing (1912-1954) i początki informatyki teoretycznej

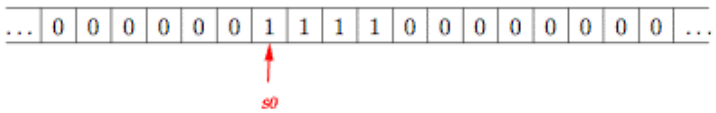
Maszyna Turinga

Składa się z nieskończonej taśmy, głowicy i programu.
Program określa, co w każdym stanie maszyna powinna zrobić w zależności od tego, co jest pod głowicą:

- zapis bitu
- przesunięcie głowicy
- przejście do innego stanu
- zatrzymanie się

Przykładowy program na maszynę Turinga. Dopisanie jedynki.





Etykiety na strzałkach określają do którego stanu przechodzimy i jaką akcję wykonuje głowica. Program z diagramu dopisuje 1 na końcu sekwencji jedynek.

Maszyna Turinga

- Turing wynalazł model niezwykle prosty, a jednocześnie równoważny obliczeniowo współczesnym komputerom: wszystko, co dzisiejsze komputery mogą obliczyć, można zaprogramować na maszynie Turinga.

Czy wszystko da się obliczyć?

- NIE! Okazuje się, że istnieją problemy nierozstrzygalne, czyli takie, dla których nie istnieje program je rozwiązujący.
- Przykłady:
  - problem stopu
  - problem odpowiedniości Posta

Problem stopu

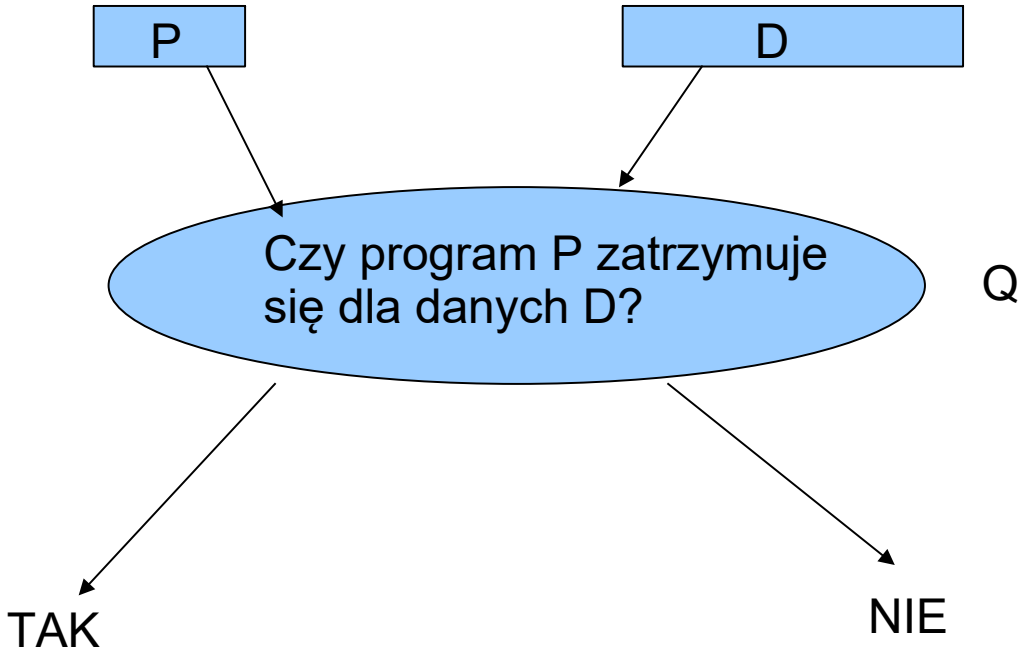
- Czy dla danego programu i dla konkretnych danych na taśmie maszyna Turinga dojdzie do stanu końcowego i się zatrzyma?
- Nieroztrzygalność tego problemu oznacza, że nie istnieje program komputerowy, który dla każdego programu P i dla każdych danych D potrafiłby stwierdzić, czy program P zatrzyma się dla danych D, czy się zapętli.

Problem stopu

- Problem stopu: napisz program Q, który dla każdego programu P i danych D rozstrzygnie, czy program P zatrzyma się dla danych D.
- Chodzi o uniwersalną metodę rozstrzygania czy istnieje niebezpieczeństwo zapętlenia się programu.

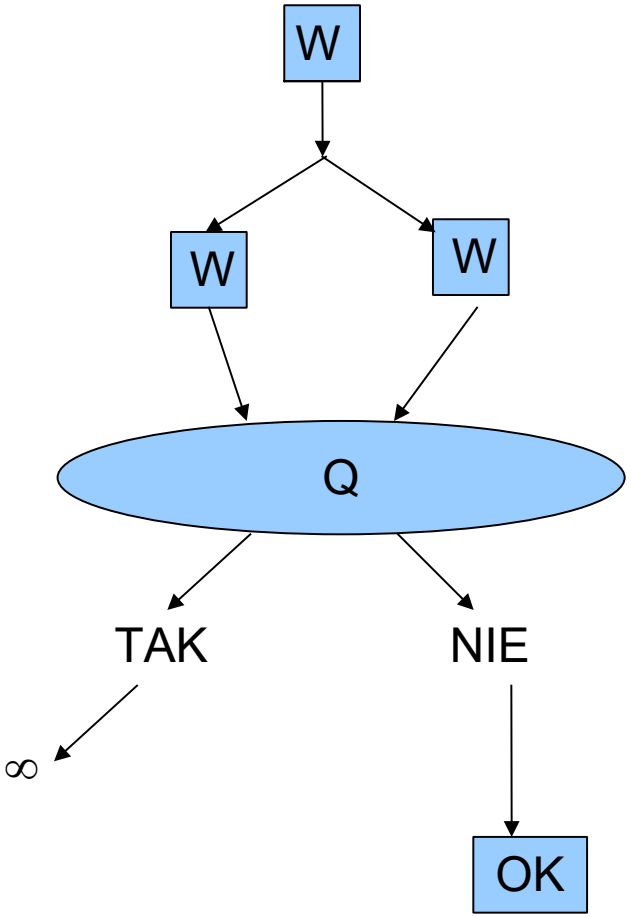
Dowód Turinga

- Chodzi o taki program Q:



Załóżmy, że taki program Q istnieje

- Konstruujemy teraz drugi program S:



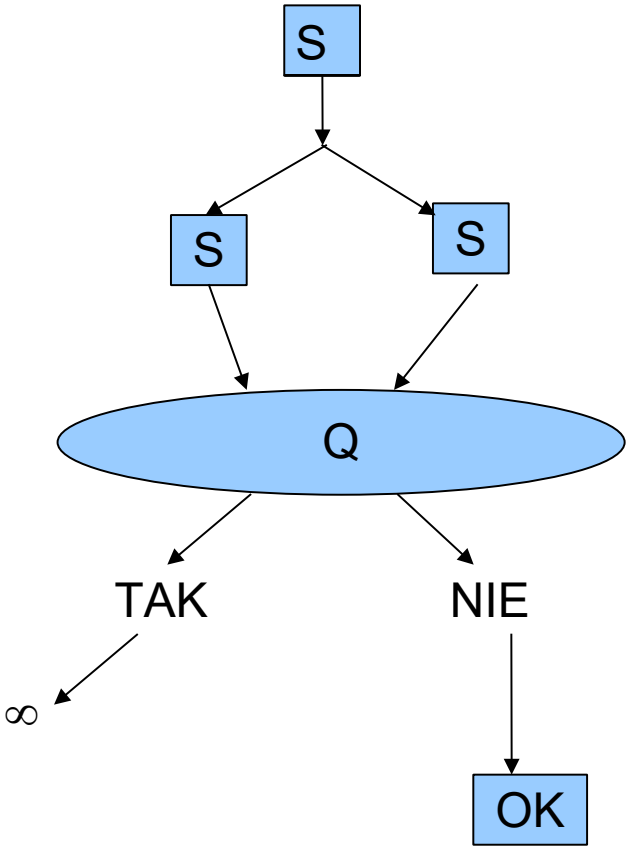
S

Jak działa program S?

- Program S ma tylko jedną daną W (jakiś program)
- Zaczyna od tego, że robi kopię tej danej
- Następnie uruchamia program Q dla programu W i kodu programu W potraktowanego jako dana.
- Jeśli program W zatrzyma się dla danej W, to program S się zapętla. Jeśli natomiast program W nie zatrzyma się dla danej W, to program S kończy działanie.

Jak działa program S dla danej S?

- Czy program S się zatrzyma dla danej S?



S

Sprzeczność!

- Okazuje się, że nie sposób odpowiedzieć na pytanie o zatrzymanie się programu S dla danej S.
  - jeśli S się zatrzymuje dla danej S, to Q da odpowiedź TAK i zgodnie z konstrukcją S, program S się zapętli
  - jeśli S się nie zatrzymuje dla danej S, to program Q da odpowiedź NIE i program S się zatrzyma.

Problem odpowiedniości Posta
Post correspondence problem

- Przykład:
  - x_1=abb y_1=a
  - x_2=b y_2=abb
  - x_3=a y_3=bb
- Czy istnieje taki ciąg indeksów i1,i2,…,in, że $x_{i1}…x_{in}=y_{i1}…y_{in}$?
- Problem odpowiedniości Posta jest w ogólnym przypadku nierozstrzygalny! Choć dla niektórych przypadków (np. dla powyższego) można podać odpowiedź, nie ma jednak ogólnego algorytmu, który dla dowolnych danych $x_1,…,x_n$ i $y_1,…,y_n$ stwierdziłby, czy można wyrównać odpowiednie słowa x-owe i y-kowe za pomocą tego samego ciągu indeksów.

Pierwsze prawdziwe komputery

Twórcy pierwszych komputerów

- Konrad Zuse
- John Atanasoff
- Howard Aiken
- John Mauchly & J. Presper Eckert

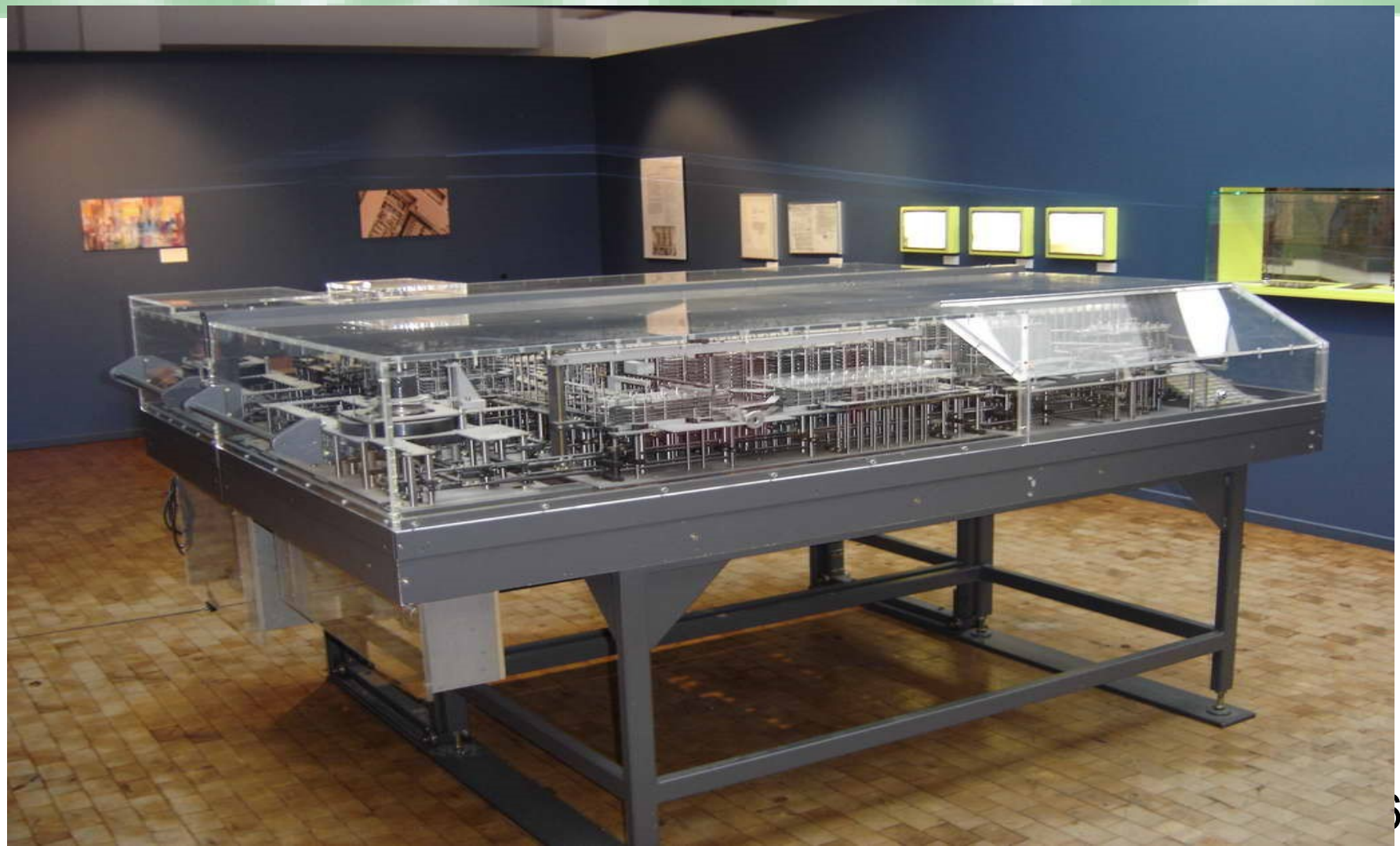
Konrad Zuse

Niemcy, 1939

“Byłem zbyt leniwy, aby trudzić sie obliczeniami, więc wymyśliłem komputer.”

Komputer Zusego działał za pomocą elektrycznych przełączników, zastąpionych później lampami.

Z1 – replika z Deutsches Museum (Monachium)

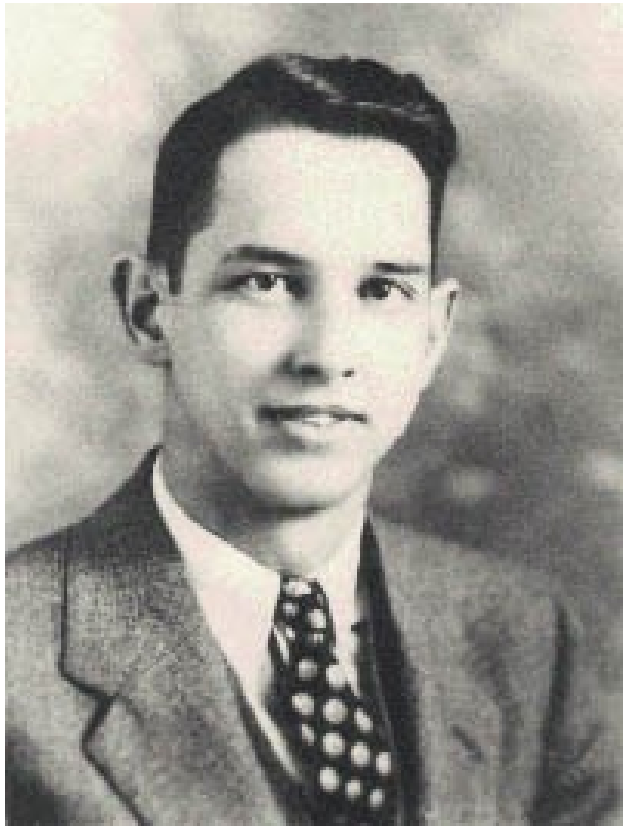


John Atanasoff

USA, 1939

Atanasoff-Berry Computer (ABC)
Komputer ABC był już w technologii lampowej i działał w arytmetyce binarnej

Nigdy nie został skończony.

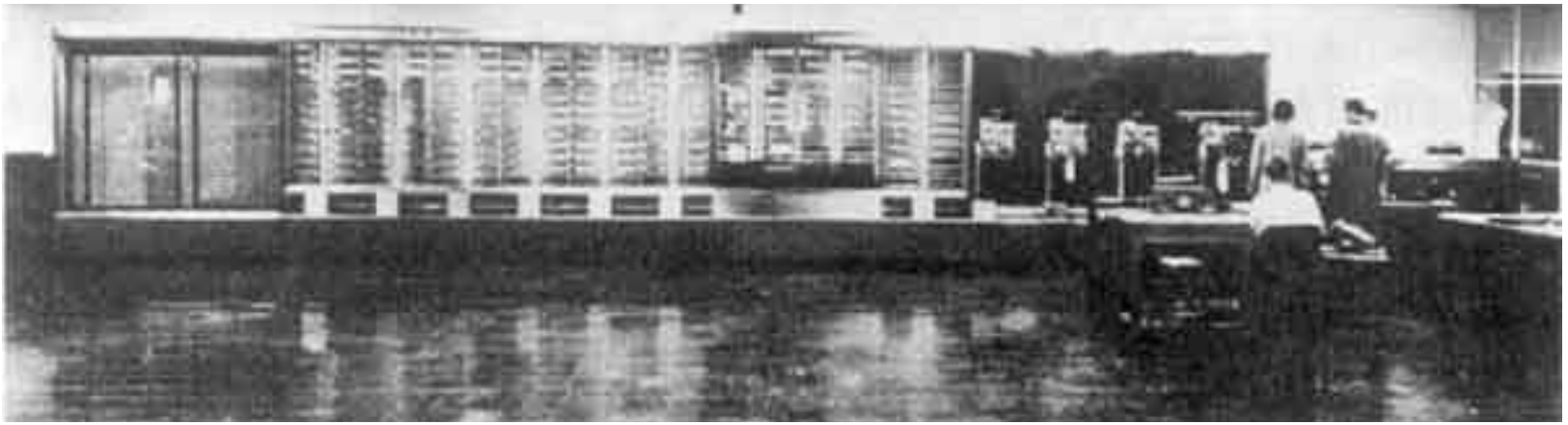


Howard Aiken

USA, 1944

Mark 1 byl największym komputerem który kiedykolwiek został zbudowany!

Działał w technologii przełącznikowej i przyjmował instrukcje wprowadzane za pomocą taśmy perforowanej.



John Mauchly oraz Presper Eckert

USA, 1945

Mauchly oraz Eckert zbudowali ENIACa (Electronic Numerical Integrator and Computer).

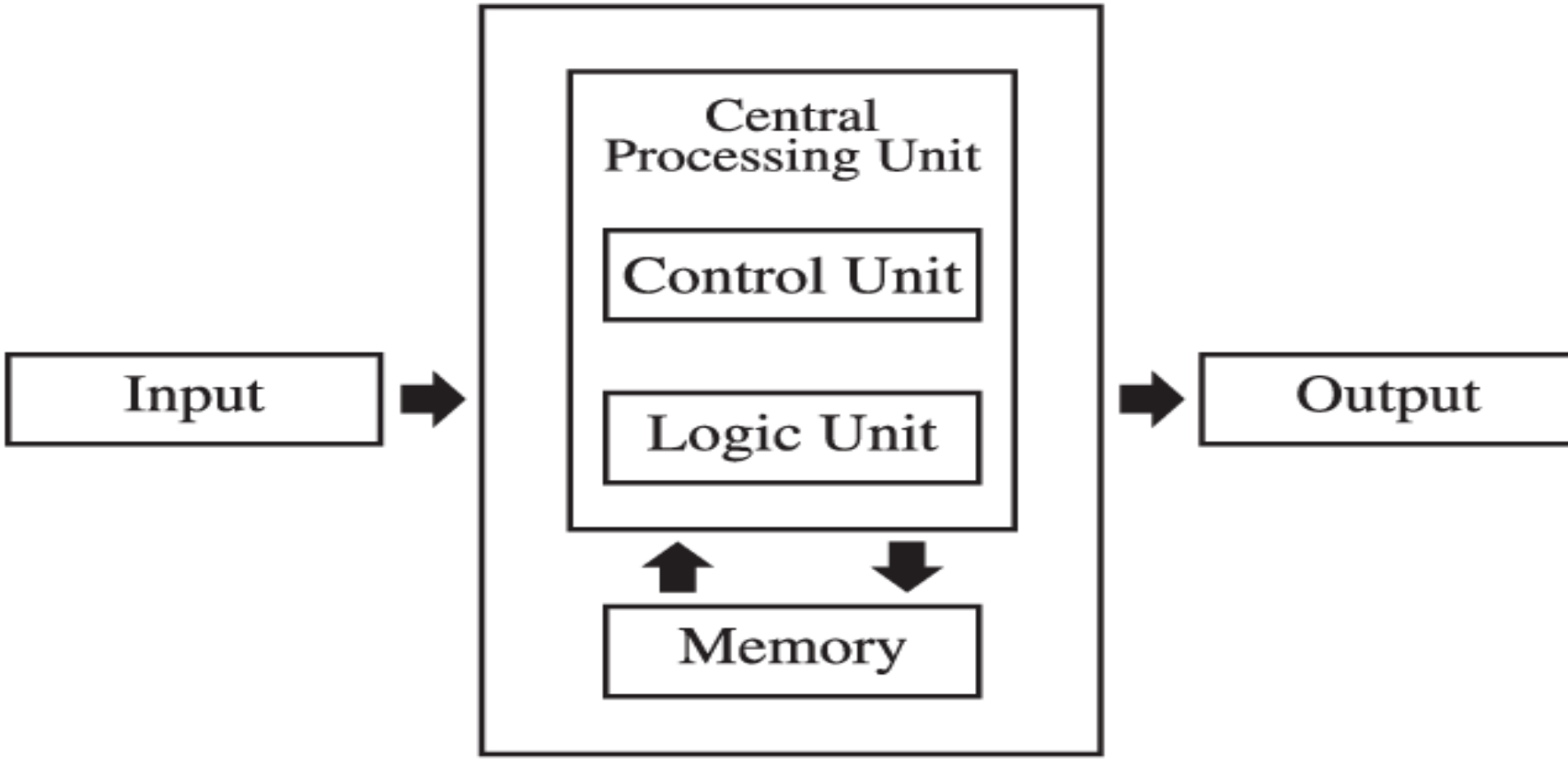
ENIAC był zbudowany w technologii lampowej i programowany za pomocą zestawiania obwodów.

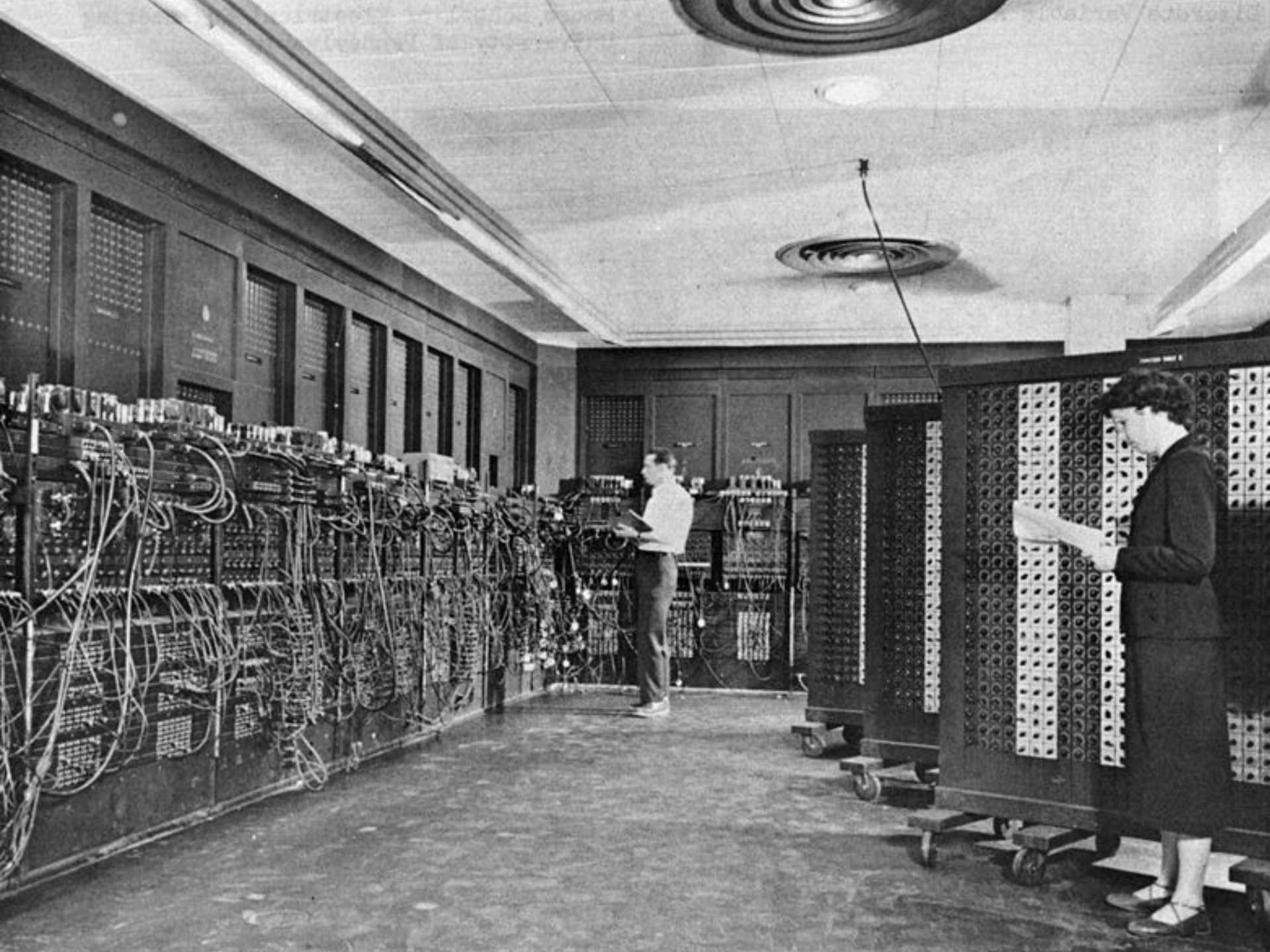
John von Neumann

Współpracując z Mauchlym I Eckertem zaprojektował sposób działania procesora (cykl von Neumanna):

-wczytanie rozkazu
-interpretacja
-wykonanie
-aktualizacja licznika rozkazów

Schemat działania komputera



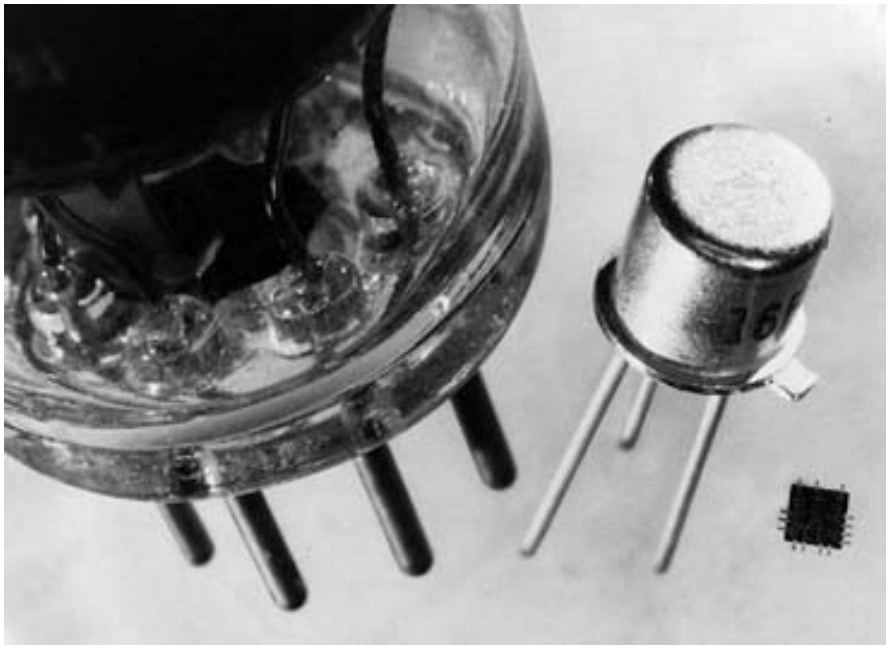


John Backus (1924-2007)

- FORTRAN (1955) był pierwszym językiem programowania z prawdziwego zdarzenia
- Miał zmienne, symboliczne adresy, łatwe zapisywanie wyrażeń,instrukcje warunkowe, pętle, procedury, ciekawe zarządzanie pamięcią

Rozwój i przyspieszenie

- Pierwsza generacja
  - Lampy próżniowe
- Druga generacja
  - Tranzystory
- Trzecia generacaja
  - Układy scalone
- Czwarta generacja
  - Mikroprocessory



Pierwsza generacja komputerów

- Lata 30-te i 40-te
- Lampy próżniowe w roli przełączników
- Duże komputery
- Niezwykle powolne, jak na dzisiejsze standardy
- Podatne na błędy
- ABC, Mark I, ENIAC, UNIVAC i in

Druga generacja

- Lata 50-te do połowy 60-tych
- Tranzystory w roli przełączników
- Znacznie mniejsze niż lampowe
- Mniej więcej tysiąckrotnie szybsze
- Tańsze i pewniejsze

Trzecia generacja

- Późne lata 60-te
- Krzemowe czipy w roli przełączników
- Znaczne obniżenie kosztu i rozmiarów
- Istotny wzrost szybkości i wydajności

Czwarta generacja

- Lata 70-te do dziś
- Zestawy przełączników zastąpione jednym mikroprocesorem
- Cena tak spadła, że stały się dostępne powszechnie